

JAERI - M
92-198

原子力知能化システム技術の研究
(人間動作シミュレーション・プログラム：HASP)
—平成3年度作業報告—

1993年1月

秋元 正幸・樋口 健二・藤井 実
久米 悦雄・神林 奨・海老原健一

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division
Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-
mura, Naka-gun, Ibaraki-ken 319-11, Japan.

©Japan Atomic Energy Research Institute, 1992

編集兼発行 日本原子力研究所
印 刷 いばらき印刷㈱

原子力知能化システム技術の研究
(人間動作シミュレーション・プログラム：HASP)
—平成3年度作業報告—

日本原子力研究所東海研究所情報システムセンター

秋元 正幸・樋口 健二・藤井 実
久米 悦雄・神林 奨・海老原健一

(1992年11月24日受理)

日本原子力研究所は、1987年からHASP(Human Acts Simulation Program)と名付けた人工知能とロボティクスに関する研究を10年計画で開始した。これは、知能ロボット、知能化プラントの基盤技術を研究開発するものである。その内容は、命令理解システムの試作、視覚認識、2足歩行ロボットの動作シミュレーション、施設形状データの作成、被曝線量計算、さらに被曝線量計算等のモンテカルロ計算の高速化を目的としたモンテカルロ計算装置の設計・試作などである。

本報告は、平成3年度のHASPの作業内容について記述する。

A Study on Intelligent Nuclear Systems
(HASP: Human Acts Simulation Program)
- Progress Report 1991 -

Masayuki AKIMOTO, Kenji HIGUCHI, Minoru FUJII, Etsuo KUME
Shaw KAMBAYASHI and Ken-ichi EBIHARA

Computing and Information Systems Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received November 24, 1992)

The fifth year progress of the Human Acts Simulation Program, HASP in short, has been presented in this report. The HASP started in 1987 at JAERI as ten-year research and development program of underlying technologies for intelligent robots, intelligent nuclear plants and so on. It consists of the research and development of technologies of a knowledge-base, robot vision, robot kinematics/kinetics, plant geometry data-base, dose evaluation and high speed Monte Carlo machine.

Keywords: Artificial Intelligence, Natural Language Processing,
Knowledge-base, Pattern Recognition, Robotics, Graphics,
Monte Carlo Method, Supercomputer, Computer Simulation

目 次

1. はじめに	1
1.1 HASPの研究概要	1
1.2 平成2年度までの経緯	2
1.3 平成3年度の研究概要	3
1.4 今後の計画	4
2. 命令理解システム	7
2.1 はじめに	7
2.2 知能構造モデル	8
2.3 問題解決手法と状態管理メモリ	8
2.4 命令理解シミュレーションシステムの評価	12
2.5 原子力施設環境用デルタゴールの推定	14
3. 視覚認識の研究	20
3.1 はじめに	20
3.2 2段階動的計画法モデルによる3次元空間認識	21
3.3 工具認識用画像データベースの試作	27
3.4 今後の研究方向	28
4. 人間形ロボットの2足歩行シミュレーション	43
4.1 はじめに	43
4.2 2足歩行ロボット・モデル	43
4.3 シミュレーション結果	50
4.4 おわりに	53
5. 施設形状データベース	72
5.1 概 要	72
5.2 旧施設形状DBにおける形状データの構造	72
5.3 形状データの変換作業	73
5.4 JRR-3メンテナンス用機器データベースの試作	73
5.5 平成4年度の計画	74
6. 被曝線量評価	95
6.1 はじめに	95
6.2 計算プログラム	95
6.3 計算モデル	95
6.4 モンテカルロ・コード用可視システムVSMCコード	97

6.5	計算結果	98
6.6	今後の課題	98
7.	モンテカルロ計算装置	105
7.1	MORSE コードのベクトル/並列化作業	105
7.2	VIMコードのベクトル/並列化作業	106
7.3	MCNPコードのベクトル化作業	107
7.4	GRAPE コードの並列化作業	109
8.	おわりに	118
謝	辞	118

Contents

1. Introduction	1
1.1 Outline of HASP	1
1.2 Progress of HASP	2
1.3 Summary of Research in 1991 Fiscal Year	3
1.4 Future Plan	4
2. Instruction Understanding System	7
2.1 Introduction	7
2.2 Model for the Knowledge Structure	8
2.3 Problem Solving and State-management Memory	8
2.4 Evaluation of the Prototype of the Instruction Understanding System	12
2.5 Estimation of Delta-goals for the Nuclear Plants	14
3. Robot Vision	20
3.1 Introduction	20
3.2 Three Dimensional Space Recognition with Two-stage Dynamic Programming Model	21
3.3 A Trial of Making Image Database for the Recognition of Tools	27
3.4 Future Research Plan	28
4. Numerical Simulation of Biped Locomotion Robot	43
4.1 Introduction	43
4.2 Simulation Model of Biped Locomotion Robot	43
4.3 Results of the Simulation	50
4.4 Concluding Remarks	53
5. Plant Geometry Data-base System	72
5.1 Outline	72
5.2 Outline of the Previous Plant Geometry D.B.	72
5.3 Transformation of Geometric Data	73
5.4 A Prototype Data-base for the Maintenance Equipments of JRR-3	73
5.5 Future Plan	74
6. Dose Evaluation	95
6.1 Introduction	95
6.2 The Calculation Program	95
6.3 The Calculation Model	95

6.4	The Visualization System For Monte Carlo Program	
	- VSMC Code -	97
6.5	The Calculation Results	98
6.6	Future Problems	98
7.	Monte Carlo Machine	105
7.1	Vector-Parallelization of MORSE Code	105
7.2	Vector-Parallelization of VIM Code	106
7.3	Vectorization of MCNP Code	107
7.4	Parallelization of GRAPE Code	109
8.	Concluding Remarks	118
	Acknowledgement	118

1. はじめに

昭和62年度に原子力委員会が改訂した「原子力開発長期計画」においては、原子力技術開発の共通基盤ともなり、かつ他の研究開発分野へも波及効果をもたらす可能性のある技術開発に取り組むことが定められた。それらの分野として、原子力用材料、原子力用レーザー、原子力用人工知能及び放射線リスク評価・低減化技術が挙げられ、これらの4分野について、昭和63年度から関連機関によって研究を開始した。原子力用人工知能については、日本原子力研究所（以下、原研）が既に開始していた「人間動作シミュレーション技術の研究」に、「極限環境センシング・システムの開発」を加え、新しい長期計画の下に研究開発を進めてきた。

本報告は、「人間動作シミュレーション技術の研究」に関する平成3年度作業報告である。また、この研究はHASP(Human Acts Simulation Program)と名付けている。

さて、平成3年度の成果報告を述べる前に、HASPの研究目的など研究概要、平成2年度までの研究概要、ついで平成3年度の成果概要、今後の計画を述べる。

1.1 HASPの研究概要

HASPの概念は、Fig. 1.1に示すように、自然文で書かれた命令を、コンピュータのなかで作られた模擬人間、即ち、ソフト知能ロボットに与え、知能ロボットは命令を読み、意味を理解し、自己の行動を計画し、動作列を生成する（命令理解、静的経路探索、ロボット作業計画）。次いで、環境の状況を認識・判断しつつ動作列を精密化して目的の位置に、ロボット動力学／運動学に基づいて移動し（動的経路探索、視覚認識、ロボット動力学）、点検作業の対象となる装置や機器が有する情報を収集しながら（センサー融合）、それが正常かどうか判断するなど点検作業を遂行する（ロボットによる異常診断、ロボットによる修復作業）。この間ロボットが受ける放射線被曝線量計算を含め、これらすべての過程は論理、数値モデルによってコンピュータ・シミュレーションされる。

以上述べたように、HASPの研究目的は、人間理解や人間そのものの動作シミュレーションではない。人的作業の定性的・定量的評価、人的作業の機械化、複雑人工構造物と人間の役割分担の明確化である。具体的には、高放射線下で動作する知能ロボット設計のための要素技術を開発すること、ロボット知能化の程度との関連において、プラントの知能化・自動化技術を開発すること、人工知能関連のシステム化された基盤技術を原子力分野の研究者、技術者に提供することがHASPの目的とするところである。

1.2 平成2年度までの経緯

HASPはTable 1.1に示すように、10年計画で研究を開始し、平成3年度で5年目になる。当初の2年間（昭和62～63年度）では、研究の道具となるワークステーション、映像生成用並列コンピュータなどのハードウェアの整備、日本語解析ソフト、ロボットの動作空間を与える原子炉施設の形状データベースなどソフトウェアを整備した。

平成元年度から2年度には、研究概要で述べた各要素技術の研究開発を進めた。即ち、命令理解システムについては、日本語で書かれた作業指示文の構文意味解析を行い、結果をフレーム構造で表現し、これを基に、行動目標（ゴール）を生成するシステムの試作に着手し、仮定された人間の知識構造に対応して、プログラムの構造化を進めた。一方、知能ロボットが行う装置・機器等の異常診断及びその修復作業に定性推論応用の検討、R. Shankらの概念依存関係を拡張し、知能ロボット動作、作業計画及びその評価への応用の検討、さらにロボット移動の経路探索へ真理管理システム（Truth Maintenance System）の応用の検討を行った。

また、視覚認識の研究については、ハードウェアとして既に開発済みの移動ロボットの環境認識技術の現状調査を行うとともに、マーク検出によるロボットの自己位置認識算定プログラムの開発、ニューラル・ネットワークの一種であるHopfieldモデルによる両眼立体視対応問題の解法の検討等を行った。

2足歩行ロボットの動力的シミュレーションについては、ユーグスラビアのヴコブラトビッチ（M. Vukobratovic）による人間型2足歩行ロボット・モデルを拡張・設備し、安定歩行を保持しつつ、ロボットの発進・定常歩行・停止の完全歩行のシミュレーションするソフトウェアを完成させた。また、シミュレーション結果の動画的も可能にした。

施設形状データベースについては、知能ロボットの動作空間となる原子力施設として、原研の研究用原子炉JRR-3の一部を、施設の設計図面からCSG(Constructive Solid Geometry)手法によってデータベース化した。データベース化に当たっては、比較的放射線レベルが高く、しかも巡回・点検時に作業員が頻繁に立入る場所を主な対象とし、2足歩行知能ロボットの動作環境として、カラー動画化を行った。

被曝線量評価については、2足歩行ロボットに要求される耐放射性を評価することがその研究の目的であるが、現在そのモデルが確定していないため、米国オークリッジ研究所において開発された人体モデル：Cristy phantomを基に、モンテカルロ・コード（MCNP: Monte Carlo Neutron and Photon Transport Code System）を用いて、ガンマ線の被曝線量計算を行った。この種の計算で問題になるのは、複雑な人体モデルの形状データの作成やその一貫性を確認するために、相当な作業量が必要になることである。これを軽減する観点から、人体モデル・データの可視化プログラムを作成した。

超高速モンテカルロ装置については、昭和63年度に概念設計、平成元年度に詳細設計を完了させ、平成2年度から3年計画で単一プロセッサの製作を開始した。また、平成3年度にはプロセッサ多重化の製作に着手し、装置全体の完成は平成5年度の予定である。本装置の設計基準として、

代表的なモンテカルロ・コードKENO-IVの演算速度が、単一プロセッサで、そのプロセッサのスカラ計算の2.5倍以上、プロセッサの多重化により、スカラ計算の10倍以上と設定している。

以上が平成2年度までの研究開発の経過であるが、平成元年度から、HASP研究の一部は、科学技術庁原子力基盤技術開発クロスオーバー研究「原子力用人工知能を具備した原子力施設のシステム評価研究」としても位置付けられている。このクロスオーバー研究では、自律型プラントに関する要素技術開発が目標として設定され、参加機関は、原研のほか、動燃、理研、電総研、船研である。原研の担当研究テーマは「シミュレーション・システムによる智能ロボットの自律型プラントへの適用性評価」であり、上記の開発済ソフトの性能評価をとおして、プラントへの適用性を明らかにしていくこととしている。

1.3 平成3年度の研究概要

詳細は、第2章以降に述べるとおりであるが、ここではその概要を述べる。

(1) 命令理解システム

平成2年度に行ったプログラムの構造化を一層進め、プロトタイプ・システムとして試作を完了させた。特徴として、仮定された人間知識構造に対応して、推論情報の一括管理、非同期な言語情報の入力や動的環境変化に対応可能な割込み命令を許容する制御構造等が挙げられる。

(2) 視覚認識の研究

左右両眼に映る画素の対応付けが可能であれば、対応付け後その視差から三角測量の原理を用いて、距離情報、即ち3次元情報が得られる。この両眼立体視の対応問題で、左右走査線上で見える順序が逆転する物体を含むシーンの対応付けが、従来手法では解かれていない。これに対し、2段階動的計画法と称する新しい手法を開発し、問題を解決することに成功した。

(3) 人間形ロボットの2足歩行シミュレーション

完全歩行シミュレーション・ソフトをさらに発展させた。即ち、2足歩行ロボット設計の観点から、与えられた歩容に対し、安定化に要求される補償動作の計算、逆に、与えられた可能な補償動作に対して、安定性のために実現すべき歩容の計算が可能になるよう機能を拡充した。この改良ソフトにより、種々の歩行安定性を検討した。

(4) 施設形状データベース

容易な記号処理を意図し、開発済データベースでは、LISP及びC言語を用いて記述したが、環境認識等で今後必要になる大規模の高速数値処理には、FORTRAN言語が適している。用途に応じたデータベース構築の観点から、現データベースの構造等の変換を行った。

(5) 被曝線量評価

人間形ロボットに対する被曝線量評価手法の妥当性を確認し手法を確立するため、人体モデル：Cristy Phantomに対し、モンテカルロ計算コードMCNPを用いて、実効線量当量の計算を行った。計算結果を国際放射線防護委員会（ICRP）の基準データと比較し良好な結果を得た。

(6) モンテカルロ計算装置

代表的モンテカルロ・コード KENO-IV, MORSE, VIM, MCNP, GRAPEのベクトル化及び並列化のため、ソフトウェアの書き換えを行った。即ち、本装置で独自に開発した専用パイプラインである幾何形状、事象分類、領域判定パイプライン意識した詳細な書き換えである。一部のコードについては、本装置での達成予想演算速度を評価した。

1.4 今後の計画

平成4年度以降の研究計画はTable 1.1に示すとおりである。命令理解システムについては、実用性の観点から、完成したプロトタイプのパフォーマンス評価を行い、問題点を明らかにする。環境認識技術の研究では、視覚認識の基本技術として両眼立体視の研究を継続するとともに、実用性を意識したロボット視覚認識ソフトの開発を進める。同じく実用性の観点から、2足歩行ロボットに限定せず、各種移動のロボットに搭載可能なセンシング・システムの開発として、視覚、聴覚、臭覚等のセンサー・ハードウェアの設計・製作及びこれによって収集されるデータを知的に処理し、環境を認識するシステムの研究開発に重点を置く。また、独自の2足歩行ロボット開発を目指し、2足歩行シミュレーション・ソフトにより設計研究を継続実施する。

Table 1.1 人間動作多体シミュレーション技術の研究

研究項目	昭和63	平成1	平成2	平成3	平成4	平成5	平成6	平成7	平成8
①人間動作シミュレーション技術の研究 ②原子力用人工知能を具備した原子力施設のシステム評価研究 (クロスオーバー研究)	構文解析								
	命令理解システムの試作								
(1) 命令理解、行動計画の研究	施設形状DBの試作								
(2) 環境設定、映像化技術の研究									
(3) 環境認識技術の研究	技術調査								
	自律型プラント設計の技術調査								
(4) ロボット動作の研究	ニューロ・モデル								
	マーク検出								
(5) ロボット・ハードウェアの試作									
	二足歩行シミュレーション用ソフトの開発								
(6) 被曝線量計算の研究	技術調査								
	人体モデルの開発								
(7) 超高速モンテカルロ計算装置の開発	概念設計								
	モンテカルロ計算コードへの適応性評価と利用技術の開発 (クロスオーバー)								

注) 上記の①は、主にモデル、手法等の基礎的研究開発である。②は、①の成果を原研の研究用原子炉JRR-3に適用し、評価するクロスオーバー研究である。

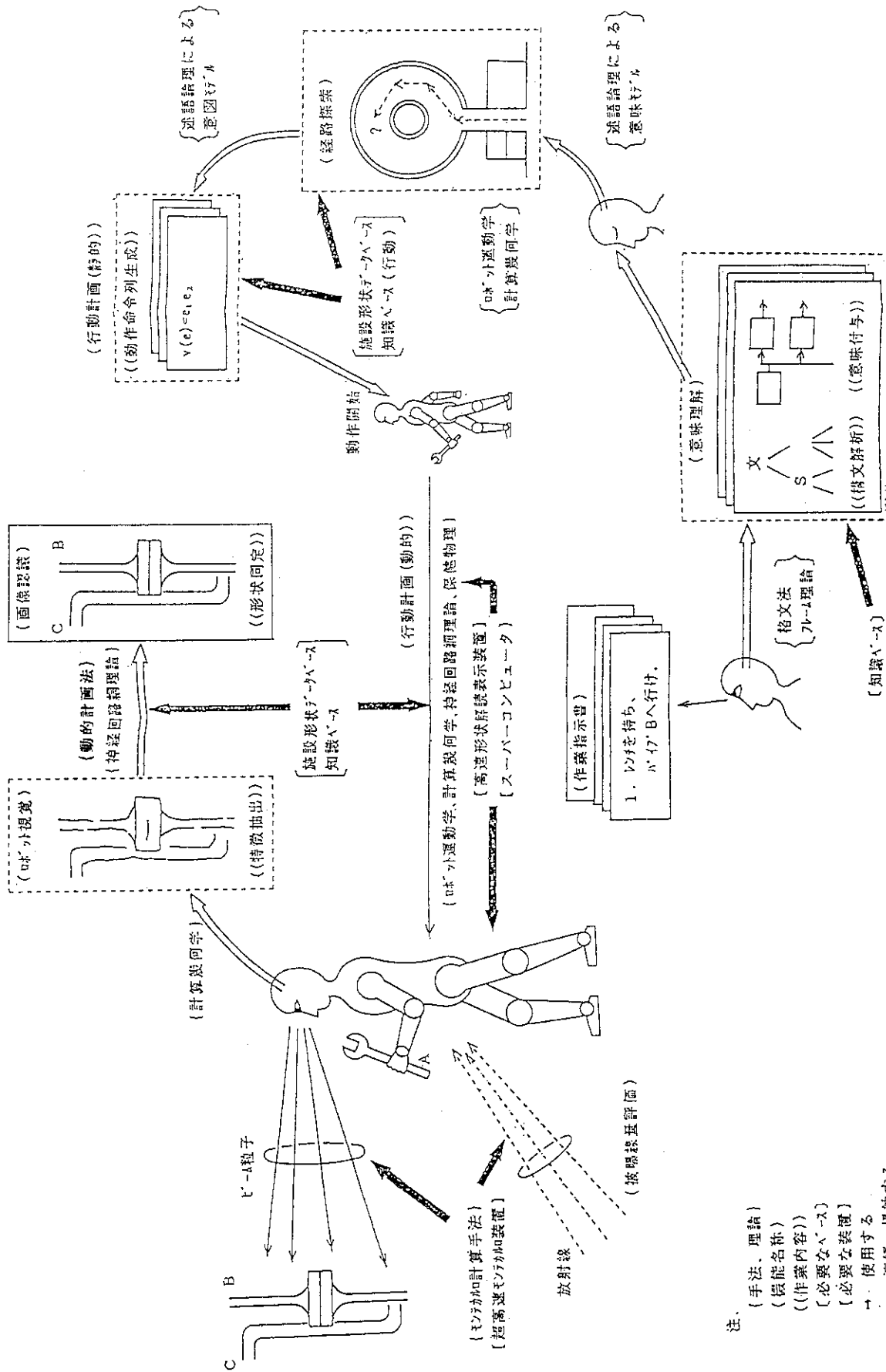


Fig. 1.1 人間動作シミュレーションHASPの概念図

注.
 (手法、理論)
 (機能名称)
 ((作業内容))
 [必要なベース]
 [必要な装置]
 ⇒ 使用する
 ⇨ 演繹、帰納する

2. 命令理解システム

2.1 はじめに

自然言語を用いて人間と対話することができ、しかも、さまざまな作業を遂行することのできる知能ロボットの実現は、あらゆる分野において切望されている。このような知能ロボットを実現するためには、自然言語理解と問題解決の2つの要素技術が不可欠である。ここで、自然言語理解の能力とは、日本語などの自然言語で記述された文（文字列）から、ロボットの行動目標を計算機で取り扱うことのできるデータ構造に変換する能力である。また、問題解決の能力とは、自然言語理解によって得られた行動目標から、ロボットの実行する具体的な動作内容を決定する能力である。HASPでは、原子力施設内で点検作業する知能ロボットに自然言語理解と問題解決の能力を与えるため、この両者を計算プログラムの形で実現する命令理解システムの研究・開発に取り組んでいる。この研究は昭和62年度から着手され、自然言語理解及び問題解決の各種手法の調査[1]、そして、関連ソフトウェアの導入・整備を行い[2, 3, 4]、平成3年度に、原研の研究用原子炉JRR-3の1部を簡略化した仮想的な世界において、Table 2.1に示した日本語命令に基づいた知能ロボットの行動をシミュレートする命令理解シミュレーションシステムの試作を完了した[5, 6]。

命令理解シミュレーションシステムはKyoto Common Lisp言語を用いSUN3/260C上で構築され、以下の特徴を持っている：

- ・人間の知能構造を模倣するために、推論情報を一括管理し、そして、無意識的なサブプロセスの概念を導入した。
- ・非同期的な言語情報の入力や動的な環境の変化、すなわち、割り込み命令に対応可能な制御構造として、認識-判断-実行サイクルを推論エンジンに導入した。
- ・自然言語の意味表現方法として、高木・伊東 [7]の手法による構文および意味的な係り受け構造を用いた。文字列として与えられた日本語文の意味構造を導出するソフトウェアとして、(株)CSKで開発されたCS-PARSERを利用した[3]。
- ・問題解決手法としてR. C. Schankらの提唱する概念依存性理論(Conceptual Dependency Theory, [8])に基づくゴールプランの理論[8, 9]を用いた。
- ・ソフトウェアの拡張性や保守性を高めるプログラミングを採用した。
- ・マルチウインドウ環境を利用して、ロボットの移動状況と問題解決のための推論状況を同時に表示可能とした。

以下では、まず、命令理解シミュレーションシステムで用いている知能構造モデルおよび問題解決手法について述べ、次いで、命令理解シミュレーションシステムの評価と今後の課題についてまとめる。

2.2 知能構造モデル

命令理解シミュレーションでは、Fig. 2.1 に示すように、(1)認識-判断-実行サイクル、(2)短期記憶と長期記憶、そして、(3)エージェント、という3つの概念を構成要素とした簡単な知能構造モデルを用いている。

行動手順の設定に際して、人間はまず大まかな手順を作り、さらに外部環境を参照しながら具体的な手順へと展開していると考えられる。移動中の予期せぬ障害物のように突発的な環境変化に対応して行動手順を修正する際、外部環境の参照は極めて重要となる。認識-判断-実行サイクルは、このような非同期的な割り込み情報に随時対処できる人間の行動制御構造を模倣したものであり、以下の処理を行う：

- | | |
|-----|---|
| 認 識 | 外部環境から得られる視覚情報や言語情報を比較的短期間保持される記憶領域（短期記憶）に取り込む。 |
| 判 断 | 短期記憶の内容と長期記憶の内容を参照して、何をなすべきかを決定する。 |
| 実 行 | 判断された動作を実行する。この作用は、外界のみに作用するとは限らず、記憶の内容を変化させることもある。 |

ここで、認識-判断-実行サイクルを1巡回するタイミングは、実行部分の処理内容に依存するものであるが、おおむね1秒程度と設定している。また、短期記憶は一過性の情報を保持する記憶の場であり、そして、長期記憶は半永久的な情報を保持する記憶の場である。認識-判断-実行サイクルの実行部分では、エージェントの概念を用いる。エージェントとは、人間が無意識のうちにやっている、ある程度まとまった処理を行う処理実行単位である。例えば、人間が自国語の言語情報を見たり聴いたりした場合、特に文法を意識しなくても文の解釈が可能である。このようなことをモデル化するには、人間の意識の外に言語を解釈する機能を持ったエージェントを仮定し、生の言語情報をエージェントに渡し、処理結果を受け取るというモデルを想定することが自然である。

上に述べた知能構造モデルに対応して、命令理解シミュレーションシステムは、認識-判断-実行サイクルを模した推論エンジンを中心に、短期記憶に相当する状態管理メモリ（自己状態メモリ、環境状態メモリ、推論状態メモリ）と、長期記憶に相当する知識ベース、そして、エージェントに相当するサブプロセスを配置した構造をとっている。（Fig. 2.2 参照）。また、下位中枢神経系の感覚中枢および運動中枢については省略しており、感覚器に相当する日本語入力インタフェースおよび模擬視覚ユニット、そして、効果器に相当する応答出力インターフェイスおよび動作シミュレータを設けている。

2.3 問題解決手法と状態管理メモリ

命令理解シミュレーションシステムの問題解決手法はR. C. Schankらの概念依存性理論とゴールプラン理論に基づいたものである[8, 9]。概念依存性理論では、動作 (actions) 表現と状態

(states)表現によって事象が表現され、動作表現と状態表現の間の因果関係(causation)が定義されている。動作及び状態表現は、基本動作(primitive acts)、基本状態(primitive states)、そして、格要素の間の係り受け関係によって表現される。基本動作および基本状態の代表的な例は、

基本動作 PTRANS (場所移動), ATRANS (所有移動), MTRANS (情報移動)
基本状態 LOC (場所), CONT (所有), MLOC (記憶内容)

である。ここで、概念依存性理論の特徴を説明するために、「ロボットが場所xから場所yに移動する。」という事象を考えよう。ロボットを主体として考えた動作は、

$$event_1 = (\text{PTRANS}(\text{ACTOR robot})(\text{OBJECT robot})(\text{FROM } x)(\text{TO } y)), \quad (2.1)$$

のように書き表すことができる。一方、 $event_1$ は、「ロボットの位置が場所xから場所yに変化する。」と書き表されるように、ロボットの位置状態の変化としてとらえることができる。この状態表現は、

$$event_2 = ((\text{LOC}(\text{ACTOR robot})(\text{VAL } x), \rightarrow (\text{LOC}(\text{ACTOR robot})(\text{VAL } y))), \quad (2.2)$$

となる。概念依存性理論では、 $event_1$ が $event_2$ の原因である($event_2$ が $event_1$ の結果である)という因果関係を、

$$event_1 \xrightarrow{r} event_2, \quad (2.3)$$

と表現する。この例において知能ロボットの問題解決を考えると、状態変化 $event_2$ をゴールとして、 $event_1$ なる動作を生成することと等価である。Schankらのゴールプラン理論は、LOC, CONT, MLOCといった基本状態に対して、その変化を意味するゴールを設定し、デルタゴールと名付けた。LOC, CONT, MLOCに対するデルタゴールは、それぞれ、

d-prox (移動ゴール), d-poss (所有ゴール), d-know (情報入手ゴール),

である。命令理解シミュレーションシステムでは、Schankらのデルタゴールの概念を参考に、以下のデルタゴールを設定した。

d-prox, d-poss, d-state-of, d-know, d-speak.

ここで、d-state-ofは、知能ロボットの操作対象となる機器(ドアなど)の状態変化を表すデルタゴールであり、極めて汎用性の高いものである。命令文入力から問題解決への流れは、以下の通りである：

- i. 入力された文字列(命令)に対してCS-PARSERを使って構文・意味解析し、フレーム形式の意味構造を生成する。
- ii. 名詞概念知識(Fig. 2.2)を参照することによって意味構造に含まれる実体概念(物体)を特定し、行動目標を生成する。
- iii. 行動目標に含まれる動詞をキーとして、動詞概念知識(Fig. 2.2)を検索し、行動目標を複

数のデルタゴールに展開する。

iv. 得られたデルタゴールを実行関数と呼ばれる基本動作に展開し、順次実行する。

ここで、実行関数は智能ロボットの基本的な動作であり、Table 2.2にその類別をまとめた。なお、機器固有の操作法に関しては、デルタゴールマクロ(DO)を用いて対処している。デルタゴールマクロとは名詞概念フレームの特定のスロットにあらかじめ実行関数を記述しておき、行動目標からデルタゴールへの展開時に、そのスロットの実行関数をそのまま取り込むものである。

既に、2.2節で述べたように、命令理解シミュレーションシステムでは非同期的な割り込み命令を受け付けることを前提としている。この際、ロボットや環境の状態表現、そして、問題解決の過程で現れる行動目標、デルタゴール、実行関数を短期記憶として矛盾なく保存するために、状態管理メモリを導入した。状態管理メモリは、自己状態メモリ、環境状態メモリ、推論状態メモリに分類される。

自己状態メモリは、推論(思考)過程に関する情報を除いたロボット自身の状態に関する記憶を保持する領域であり、以下のCommon Lispの構造体によって定義される：

```
(defstruct selfmem
  (cbei nil      :type list) ; 現在の持ち物(current belongings)
  (pbel nil      :type list) ; 過去の持ち物(past belongings)
  (cpos '(20 40 0):type list) ; 現在の位置(current position)
  (ppos '(20 40 0):type list) ; 過去の位置(past position)
  (cdir 0        :type fixnum); 現在の方向(current direction)
  (pdir 0        :type fixnum); 過去の方向(past direction)
)
```

自己状態の変化を認識するために、最新の状態と、前回の認識-判断-実行サイクルの状態とを関連付けて保持しておかなければならない。そのために、持ち物、位置、方向のそれぞれについて現在の項目と過去の項目を設けている。本来ならば持ち物、位置、方向の3メンバにして、その中で分類すべきであるが、単に処理効率上の理由から6メンバの構造体になっている。

環境状態メモリはロボット自身以外のこの状態に関する記憶を保持する領域である。ここでの情報は感覚器(外界入力インタフェース)から得られるものであり、

- ・日本語入力インタフェースからの入力文文字列
- ・模擬視覚ユニットからの疑似視覚情報(視覚情報配列)
- ・LispリスナーからのLispフォーム

の3種類を想定している。環境状態メモリのデータ構造は、

```
(defstruct envmem
  (vision nil :type list) ; 疑似視覚情報(vision data)
  (lang "" :type simple-string); 入力文文字列(language data)
)
```

なる入力文と模擬視覚情報のメンバとするCommon Lisp構造体で表現している。

推論状態メモリは、問題解決に関する推論・思考の過程で形成される情報を保持する領域である。推論状態メモリ自体はプログラム上では2種類あり、推論状態メモリ本体と、推論エンジンの実行部分で実行すべき行為内容だけを一時的に保持するものがある。後者は単に推論エンジンの実行部分とのデータの受け渡しに用いるバッファに過ぎないので、以後では推論状態メモリ本体を単に推論状態メモリと呼ぶことにする。データ構造は、

```
<実行部で実行すべき行為内容> := 《S-expression》
<推論状態メモリ> := ( {<推論情報パッケージ>} * )
```

である。実行部分で実行すべき行為内容は、Lispのエバリュエータが評価できる形式に限られる。つまり、組み込み関数evalの引数となりうるフォームでなければならない。推論状態メモリ（本体）は、推論情報パッケージの任意個のリストである。

推論情報パッケージ（単にパッケージとも表す）とは、入力文、フレーム構造、ゴールなど、ある問題解決に関する情報をまとめて関連づけて記憶しておくためのデータ構造であり、割り込みによる情報の退避や復帰の処理や予約によって実行を保留される問題解決（特定の接続助詞“たら”を介する従属節によって問題解決の開始時期が指定される場合）に対して有効である。推論情報パッケージは、

```
(defstruct inferpac
  (idnum 0 :type fixnum) ;パッケージid(packet id number)
  (selfm nil :type selfmem*) ;受理時点の自己状態(self-memory)
  (sent nil :type list) ;入力文(input sentence)
  (frame nil :type list) ;フレーム構造(frame structure)
  (goal nil :type list) ;ゴール(goal)
  (cond nil :type list) ;推論実行条件(condition)
  (dgoal nil :type list) ;デルタゴール列(dgoal array)
  (dhist nil :type list) ;デルタゴール処理履歴(dgoal history)
)
```

なるCommon Lisp構造体で表現する。各メンバの内容は、以下の通りである。

- パッケージid: パッケージを特定するユニークな番号である。この値はパッケージ生成と同時に自動的に決定される。
- 受理時点の自己状態: 命令を受理した時点の自己状態メモリの内容である。例えば「測定して来い」という命令の場合、「来い」に関する移動の目的値が陽に述べられなかった場合は命令受理時点を目的地とするようにしているが、その際にこのフィールドの情報を参照する。
- 入力文: 問題解決のきっかけとなった日本語入力文文字列である。
- フレーム構造: 入力文の構文・意味解析した結果から生成したフレーム構造である。
- ゴール: フレーム構造から行動の目標として設定したゴールである。入力文が単文の場合はそのフレーム構造とほぼ同じであるが、複文の場合は文を分割して単文単位のフレームに相当す

るフレーム構造が入る。

- 推論実行条件：現時点で問題解決を行なってよいかどうかを表わす情報である。
- デルタゴール列：問題解決用個別知識を用いてゴールをデルタゴールの列に展開した結果である。
- デルタゴール処理履歴：デルタゴール列を実行関数列に展開した結果が入る。また、このフィールドは、実行関数列を処理する際のワーキングエリアを兼ね、処理の履歴を保持する領域でもある。

ここで、推論エンジンによる推論状態メモリの制御方法について簡単にまとめる。推論エンジンの判断部分では、まず環境状態メモリをチェックして、入力文文字列があるならば、「その入力文をCS-PARSER に渡して処理する」という実行内容を設定する。この結果新しい推論情報パッケージが生成される。入力文文字列が無ければ、優先度の高い推論パッケージを選択し、そのパッケージに対して実行部分で行う処理内容を決定する。推論状態メモリは推論情報パッケージのリストであり、基本的にはLIFO(Last in First out)バッファとして機能する。必要であれば、パッケージの位置を移動させることによって優先順位を変更することも可能である。

2.4 命令理解シミュレーションシステムの評価

命令理解シミュレーションシステムの成果としては、まず、概念設計の段階で人間の知能構造を簡単にではあるがモデル化して、それに基づいてシステムを開発したことにより、拡張性が高く、直感的に自然な方法で問題解決を行う枠組みを確立できた点が挙げられる。ここ数年における知能ロボットの研究においては、自然言語理解と問題解決に的を絞って検討を重ねてきた。しかし、人間の知能構造は決して自然言語理解や問題解決だけのためのものではなく、動作制御等も含めて知的活動全体を司るものである。自然言語理解と問題解決だけを考慮した制御構造では、少しでも異質な機能を追加しようとするとなりに行き詰まりがちであり、制御構造のハードウェアへの実装に際しては無理が生じると思われる。今回採用した認識-判断-実行サイクルを取り入れた制御構造であれば、今年度は検討しなかった実行時のフィードバック制御等も比較的容易に取り入れることが可能と考えられる。また、構造を持った記憶の一つの表現方法として推論情報パッケージを考案し、これの導入によって、推論情報を一括管理することができるようになり、問題（あるいは行動目標）が一つであっても複数個であっても安定して対処できるようになった。更に、このことによって、限定的ではあるが副作用のある割込命令からの復帰を実現することができた。システムの外見上の成果としては、実行速度が速くなり、少なくともデモにおいては、待たされているという感覚は起こらない程度の速度を実現した。

現在までのところ、自然言語理解と問題解決に重点をおいた研究であったため、ロボットの動作制御の面で問題点が比較的多い：

判断部分：本システムでは、実行関数の実行の効果は必ず期待した通りになるという仮定に基づいているので、失敗を認識したり、それをリカバーする機能は持っていない。つまり、フィード・

バックを行っていないので、このままの方式でロボットに実装することはできない。但し、移動できない場所への移動に関しては、動作シミュレーションがエラーを検出するが、いずれにしる誤動作の回復を行なうわけではない。また、副作用のある割り込みからの復帰は、ドアの開閉と移動によるものに限定しているが、これら以外の副作用については検討していないので未対応である。また、移動に関する割り込み復帰で唯一対処不能なパターン（ドアの所で「～して来い。」と命令された場合に帰ってこれない）がある。これは、移動に際して、開けたドアは必ず閉めるという知識に従って行動し、帰ってくる時にドアのある位置に移動しようとして失敗するという現象である。このようなことは、失敗のリカバー（移動先に邪魔な可動物があるときは、その可動物をどける等）ができれば解決する。現在のところ、移動（d-prox）にのみデルタゴール再分割を行なっているが、これは他のデルタゴールについても拡張する（もしくは、ゴールの階層関係を3層に固定せずに任意とする）必要があると思われる。

実行部分：判断部分で決定した行為内容を単に実行するだけであるので、それほど問題は起こらない。但し、持ち物の場所情報の補正をここで行うのは不自然であるといえる。これは、ある事象の副作用をモデル化する検討が十分でないことに起因する。本来は、ロボットの持ち物だけでなく、全ての物理的な関係を持つ物体間についていえることである。例えば、慣性の法則（水が入った鍋を動かせば中の水も同じように動く）や、エントロピー増大の法則（鍋を火にかけると中の水も温度が上がる）などの自然法則によって、常に記憶の内容を更新していくメカニズムが必要なのであろう。但し、これを実行部分の機能として持たせるべきかどうかはわからない。

推論状態メモリについては、ある程度記憶の内容を構造的に表現しているが、自己状態メモリと環境状態メモリについては、記憶の内容が離散的である：

自己状態メモリ：自己状態メモリの記憶内容について検討が不十分である。現在のところ自己状態メモリの記憶項目としては、持ち物、位置、方向の3種類を設定しているが、本来はロボットの可変的な属性のなかで直ちに認知可能なもの全てを記憶しているべきであるし、属性概念以外にも記憶しておくべきものがあるはずである。また、ある命令を受理した地点を特別に記憶するようにしている。人間ならば、場面、事象に関する情報を記憶することができ、命令受理地点を思い出す場合には、まず命令を受理した場面に関する記憶（通例、エピソード記憶といわれる）を想起し、そこから場所情報を抽出しているように思われる。このように、現在では記憶が離散的であるので、ごく単純でかつ限定的な問題にだけしか対処できない。

環境状態メモリ：環境状態メモリに関しても、自己状態メモリとほぼ同様なことがいえる。

現時点で最も問題となっているのが、知識（特に動詞概念）の表現方法である：

名詞概念知識ベース：現在のところ、名詞概念知識はフレーム構造で表現しているが、属性の設定のしかたが、やや場当たりのものである。属性には、多くの名詞概念に共通のものや、ある特定の概念に共通のものがあるが、これらの区別が必要になると思われる。現在の記述方式では、他の概念との関係の記述能力不足が認められ、オブジェクト指向を基にした記述方式の導入の検討も必要と思われる。例えば、2つの物体が接触しているだけなのか、それとも結合しているのかによって生じる物体の振る舞いの差異や、タンクの圧力という概念と圧力計の指示値という概念が実は等しい

ということなどを自然に理解するための枠組みについて検討する必要がある。また、施設形状データベースのデータ構造を変換して、名詞概念知識を構築する手法の検討が必要である。

動詞概念知識ベース：現地点では動詞概念知識が目的-手段型だけしか存在しないが、スクリプト的な知識、マニュアル的な知識のように、目的-手段型ではないタイプの動詞概念知識も保持すべきである。また、今回設定したデルタゴールだけで行動目標の記述が、一般的に可能かどうか吟味する必要があり、この点について2.5節で詳細に検討する。

2.5 原子力施設環境用デルタゴールの推定

命令理解シミュレーションシステムで使用した5種類のデルタゴールは、Schankらの提案に基づいているものの、原子力施設環境用に十分であるとは言い難い、そこで、国立国語研究所発行の分類語彙表[10]から、原子力施設環境における作業の記述のために必要な動詞概念を抽出し、その結果から、デルタゴールの推定を行った。

分類語彙表では、単語を「体の類」と「用の類」という2つに大別しており、「用の類」が動詞概念に相当する。「用の類」に掲載されている見出し数は、4739語である。その中から以下の基準にしたがって原子力施設環境用の動詞概念を抽出した。

- i. 助動詞扱い（例：られる、～じみる）：分類語彙表では用の類に含まれているものであっても、日本語解析システムCS-PARSERの品詞体系において助動詞となっているものについては、ロボット用述語動詞から除外した、ロボット用述語動詞に含まれないものは、当然ながら原子力施設環境用述語動詞とはなりえない。
- ii. 敬語表現（例：ございます、いただく）：敬語、謙讓語、丁寧語等は除外した。この中には文語的表現も含まれている。
- iii. 状態叙述（例：似る、にじむ）：ある状態を客観的な視点からみたときの表現として用いられるものや、事物の特性を陳述するものなどは除外した。但し、抽象実体主体や無意志体主体との違いはやや曖昧である。
- iv. 抽象実体主体（例：対する、わざわざする）：主語となりうる語が抽象実体概念であり、具象実体概念であるロボットが主語となりえないものは除外した。
- v. 無意志体主体（例：脱げる、沸き上がる）：主語となりうる語が無意志体主体（鉱物、工業製品等）であり、一応有意志体とされているロボットが主語となりえないものは除外した。
- vi. 生物主体（例：背負う、わずらう）：天然の生命体に固有と思われるものや、ある特定の生物の形態に依存するものは除外した。例えば、「羽ばたく」などは翼を持っている生物だけに用いられる。翼を備えたロボットも想定できなくはないが、人間型の2足歩行ロボットを前提としているので、ロボット用述語動詞からも除外してある。
- vii. 権利者主体（例：裁く、取り締まる）：人間誰もが主体となれるものではなく、主語として成り立つためにはその主体にある特定の権利を持っている必要があるものについては、ロボットがそのような権利を与えられるとは考えにくいので、これを除外した。

- viii. 殺傷行為（例：亡ぼす，枯らす）：非道徳的な行為と思われるものは除外した，もしも農業用ロボットであれば害虫を駆除するといったことも考えられるので，ロボット用述語動詞には含めてあるが，原子力施設環境用述語動詞には含まれない。
- ix. 業務外行為（例：かきまぜる，焼き払う）：原子力施設環境における巡回点検作業を想定し，そこで命令されることがないと思われるものは除外した。これらの多くは適用業務が違えば命令もありうるので，ロボット用述語動詞には含まれるが，「名乗り出る」，「踏みはずす」のように，命令してもあまり意味がないと思われるものはロボット用述語動詞にも含めていない。

抽出された原子力施設環境用述語動詞は 243語であり，この結果から，Table 2.3 に示したような原子力施設環境用に必要と考えられるデルタゴールを推定した。

サ変動詞を対象としなかったことが大きな要因であるかも知れないが，意味が明らかに異なる動詞であるにもかかわらず，同一の知識構造を持つものが予想以上に多かったことは特筆すべきである。例えば，「A地点からB地点に行く」と「A地点からB地点に帰る」では意味が異なるが，ロボットの実際の動作としては，両者ともA地点からB地点に移動するという内容になる。同じようなことが「入る」と「出る」等についても言える。これは，自然言語の動詞は主観的な見方をする場合が多く，事象としては同一でも視点によって表現を変えることが頻繁に起こるということであり，意外と動作のプリミティブはごく少数ですむのではないかという印象を受けた。動詞概念知識を静的に記述できる動詞はやはり少ないようである。特にサ変動詞を含めて考えると，単に格要素に依存するものだけではなく，総称的な動詞が多く出てくるものと思われるので，範囲を広げた調査が望まれる。格要素に依存するものについては，やはり名詞概念知識を参照して，方法に関する知識を利用するのが現時点では最善であろう。動詞概念の意味の表現もさることながら，その行為を実行する際に満たさなければならない前提条件の表現も重要である。ほとんどの場合，前提条件を事前に数え上げておくことは不可能である。しかし，人間は前提条件を全て知らなくても問題解決を行っているようである。前提条件を如何に蓄積しておくかが今後問題となるであろう。

参 考 文 献

- [1] 上中淳二, 神林 奨: 核データ評価コードに関する知識構造の調査, JAERI-M 88-143, 日本原子力研究所 (1988. 8).
- [2] 浅井 清他: 原子力知能システム技術の研究-昭和63年度作業報告-, JAERI-M 88-023, 日本原子力研究所 (1989. 3).
- [3] 神林 奨, 上中淳二: 日本語を受け付ける知識ベースシステムの試作, JAERI-M 89-218, 日本原子力研究所 (1989. 12).
- [4] 浅井 清他: 原子力知能システム技術の研究-平成元年度作業報告-, JAERI-M 89-023, 日本原子力研究所 (1989. 3).
- [5] 浅井 清他: 原子力知能システム技術の研究-平成2年度作業報告-, JAERI-M 91-101, 日本原子力研究所 (1991. 6).
- [6] K. Asai *et al.*, -A study of intelligent robots in nuclear systems-HASP: Human Acts Simulation Program, in Proc. of ICHMT 2nd International Forum on Expert Systems and Computer Simulation in Energy Engineering (Erlangen, Germany, 1992).
- [7] 高木 朗, 伊東幸弘: 自然言語の処理 (丸善, 1987).
- [8] R. C. Schank and C. K. Riesbeck, 石崎 俊監訳: 自然言語処理入門 (総研出版, 1987).
- [9] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals, and Understanding* (Lawrence Erlbaum Associates, N. J., 1977).
- [10] 国立国語研究所: 国立国語研究所資料集 6 「分類語彙表」 (秀英出版, 1964).

Table 2.1 Sample instructions for the prototype of the instruction understanding system.

例文1	PCS熱交換器付近の線量を測定して来い。
例文2	一次区画の線量測定をして下され。
例文3	PCS主ポンプの圧力をチェックしろ。
例文4	一次系の配管のリークを調べろ。
例文5	ROSタンクのレベルを調べて来い。
例文6	PCS熱交換器のメータを調べて来い。
例文7	重水区画の異常を調べて来い。
例文8	一次区画を点検して来い。

Table 2.2 Executable functions for the prototype of the instruction understanding system.

実行関数名	プログラム上での関数名
ptrans(:from :to :up-down :speed)	execf_ptrans
search(:obj :attri)	execf_search
check(:obj :attri)	execf_check
operate(:obj)	execf_operate
grasp(:obj)	execf_grasp
push(:obj)	execf_push
open(:obj)	execf_open
close(:obj)	execf_close
speak(:sentence)	execf_speak

Table 2.3 Delta-goals extracted from analyses of verbal concepts appeared in Ref.[10].

名称	意味	引数
no_operation	何もしない	
d_prox	自分が移動する	:from 起点 :to 終点
d_rotate	自分が自転する	:direction 方向
d_ptrans	対象物を移動させる	:obj 対象物 :from 起点 :to 終点
d_propel	物理的に力を加える	:obj 対象物 :means 手段 :direction 方向 :strength 強さ :complement 達成状態
d_carry	対象物を持つ (携行)	:obj 対象物
d_grasp	対象物をつかむ	:obj 対象物
d_release	対象物を放す	:obj 対象物
d_speak	発話する	:semantics 意味表現
d_know	知る	:value 値
d_watch	見る	:obj 対象物
d_listen	聞く	:obj 対象物
d_plus	加算する (a + b)	:a 数A :b 数B
d_subtract	減算する (a - b)	:a 数A :b 数B
d_multiply	乗算する (a * b)	:a 数A :b 数B
d_divide	除算する (a / b)	:a 数A :b 数B
m_exec_method	手続きを実行する	:method 手続き
m_stop_method	手続きを中断する	:method 手続き
m_abort_method	手続きを中止する	:method 手続き
m_continue_method	手続きを続ける	:method 手続き
m_repeat_method	手続きを繰り返す	:method 手続き
m_finish_method	手続きを済ます	:method 手続き

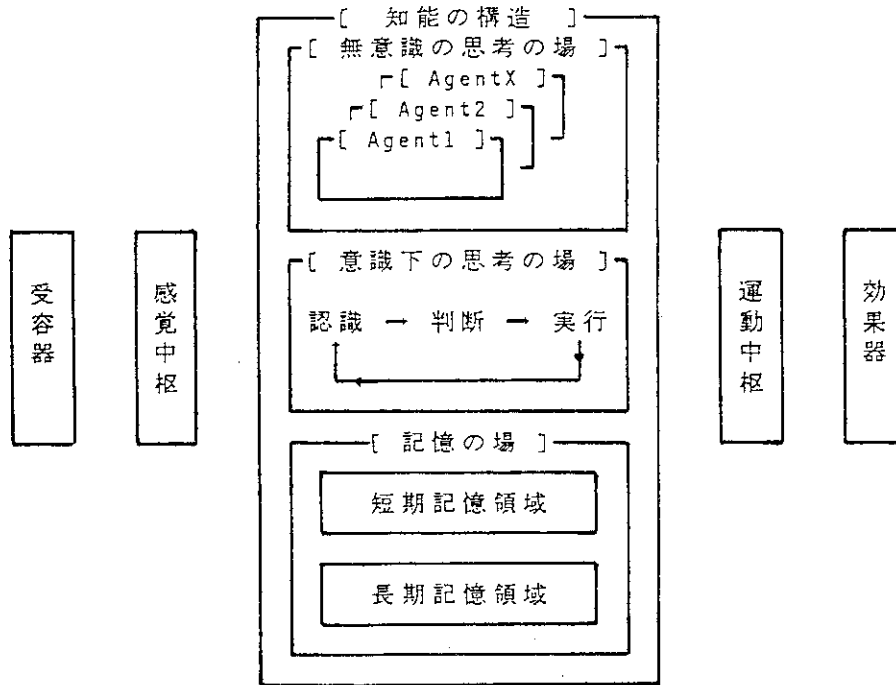


Fig. 2.1 A simplified model for the knowledge structure.

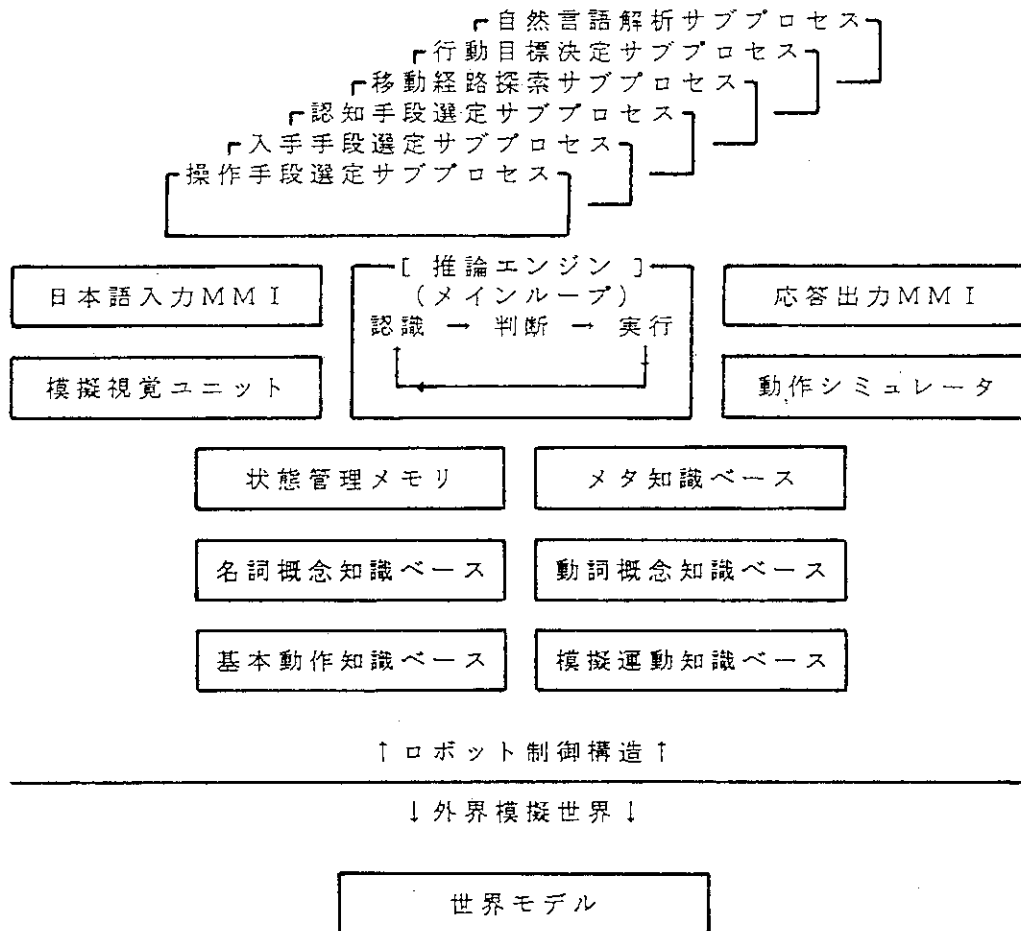


Fig. 2.2 System organization from the simplified model for the knowledge structure.

3. 視覚認識の研究

3.1 はじめに

人間動作シミュレーション技術の研究 (HASP) では、原子力施設で働く任意位置移動可能な自律型知能ロボットを研究対象とし、人間と同じ作業環境で動作することを想定して人間型の2足歩行ロボットを中心に研究開発を進めている^{1) - 5)}。

任意位置移動可能なロボットは、あらかじめ決められた軌道上を動くロボットと違い、視覚の役割が飛躍的に増大する。しかし、これまでの数多くの研究にもかかわらず、移動ロボット用の視覚認識技術は、まだ極めて貧弱である。超音波センサで前方の障害物の位置を認識する技術、単眼視画像から白線や特定のエッジを検出してナビゲーションに利用する技術、シーン中から特定の作業対象物を検出する技術等はほぼ確立されている⁶⁾が、ロボットの前方の状況(どこに何がどのような状態で存在するか)を理解する技術については、まだほとんど研究が進んでいない。

HASPにおける視覚認識の研究は、ロボットの視覚装置(ハードウェア)で収集、生成された画像情報からロボットの見ているシーンを理解、認識する過程を対象として、昭和63年秋より研究を開始した。HASPでは、既知環境下における視覚認識と半既知環境下における視覚認識について、現在それぞれFig. 3.1とFig. 3.2に示す認識方法²⁾を検討している。まず、今年度(平成3年度)の研究概要を以下に記述し、その後これまでの経緯を示す。

今年度は、両眼立体視について、前年度開発した2段階動的計画法モデルを拡張し、左右画像の画面对応問題および3眼視対応問題に適用できるようにした。その適用結果を3.2に示す。また、物体認識について、工具をいろいろな角度から見てもそれがどの工具であるかを認識できるようにするため、工具認識用画像データベースを試作した。これを3.3に示す。

[これまでの経緯]

昭和63年度は、神経回路網理論(ニューロ手法)による画像認識の可能性を調査するため次の2つの認識実験を行った。

- (1) バックプロパゲーション(BP)モデルによる手書き数字の学習実験^{1)・7)}
- (2) ネオコグニトロン・モデルにより手書き数字及び簡易画像の認識実験^{1)・7)}

両実験とも独自に処理プログラムを作成し、ドットイメージの入力データを使用して認識を行わせた。この結果、これらのニューロ手法は現段階では2値化(白黒)データで表現された単一文字または単一の簡易な図形の認識が限度で、知能ロボットの視覚装置から入力される一般画像(多数の物体が一枚の画像中に内包されている画像)の理解に直接適用することは非常に困難であることがわかった。このため、一般画像の理解には既存の画像処理手法を適用して、まず画像中の物体を抽出し、次に抽出した物体の特徴抽出を行い、ニューロ手法は抽出・分離された物体の解釈や画像中の文字の認識に使用することにした。

平成元年度は、HASPの視覚認識の研究段階を既知環境下における視覚認識、半既知環境下における視覚認識、未知環境下における視覚認識という三段階に設定し、既知環境下の視覚認識方法 (Fig. 3.1) と半既知環境下の視覚認識方法 (Fig. 3.2) について検討した³⁾。既知環境下の視覚認識では、原子力施設内にあらかじめ位置認識用マークを設置し、それをマーク検出用テンプレートを使用して画像の中から検出し、ロボットのナビゲーションに利用する方法について検討、評価した²⁾。また、半既知環境下の視覚認識では、両眼立体視対応問題の画素レベルの対応をニューロ・モデルの一種であるHopfieldモデル⁸⁾によって解く方法を検討した²⁾。

平成2年度は、既知環境下の視覚認識では、既にハードウェアが開発されている実際の室内用移動ロボットについて、それらの環境認識方法を調査した³⁾。一方、半既知環境下の視覚認識では、Fig. 3.2に示すシーン理解の最初のネックとなっている両眼立体視の研究を本格的に開始した。前年度検討したHopfieldモデルで実際に左右画像の行対応問題 (左画像の1行と右画像の1行の対応付け問題) を解いた結果、この方法では問題記述に非常に多くの変数 (対応させる1行の画素数をNとするとNの4乗個の変数) を必要とし、また、解を得るための収束計算に多くの時間がかかることがわかった。しかし、このモデルにおいて用いた、左右画素の対応問題を複数の変数で記述する方法を拡張すると、従来まだ解くことに成功していない“左右走査線上で見える順序が逆転する物体を含むシーンの対応問題 (順序逆転を含む問題)”を動的計画法で解けるようにできることに気付いた。そこで、この順序逆転を含む問題を解く2段階動的計画法モデルの開発を開始した。ただし、この手法開発は、基礎研究的色彩が強いので、原研の特別基礎研究の予算で行った。そして、順序逆転を含む問題を解く2段階動的計画法モデルの基本アルゴリズムを (行対応問題用) を開発した。

3.2 2段階動的計画法モデルによる3次元空間認識

3.2.1 はじめに

ロボットビジョンの研究は、移動ロボット研究の進展とともに、model-basedな物体認識からより広汎な未知環境の空間認識へとその中心課題が移ってきている⁹⁾。

空間の絶対距離を得る方法には、レーザー光や超音波等を用いる能動的方法と立体視等による受動的方法があるが、前者は特殊な装置を必要とし、光や波の干渉等に弱いため使用できる環境が限定される。一方、後者の代表的な方法である両眼立体視は、カメラで撮られた左右画像を正しく対応付ければ、三角測量の原理でシーンの距離情報を簡単に得ることができるので、暗闇や煙の中など一部の特殊環境を除けば、ほとんどの環境で簡単に使用できる優れた距離計測法である。

しかし、この両眼立体視も左右画像の対応付けを正確に行うことが意外に困難¹⁰⁾である。両眼立体視では、撮像時のカメラパラメータが既知の場合には、左右画像の対応付けはエピソード線と呼ばれる左右の走査線対ごとの対応点探索 (走査線内探索) に分割して解くことができる。これまで動的計画法^{11) - 15)}、相関法¹⁶⁾、弛緩法^{17), 18)}、粗密法¹⁹⁾や多眼視²⁰⁾等を用いた数多くのステレオアルゴリズムが提案されてきた²¹⁾が、Fig. 3.3に示すような左右走査線上で対応要素 (対

応点)の順序が入れ替わる対象,例えば手前に細い柱や配管,木の枝などのあるシーン,を正確に対応付けた手法はまだない。特に,大局的な対応付けを可能にする動的計画法を用いる場合は,手法の原理的制約から,“対応点の順序は入れ替わらない”とする仮定を設けざるを得なかった¹⁰⁾。これに対して,佐藤ら¹⁸⁾は,フェンス越しのシーンを対象として,対応点の順序逆転を含む場合の対応付けをHopfield型神経回路網のエネルギー最小化モデルで試みているが,順序の入れ替わりに関して特別な考慮を払っていないため,手前のフェンス付近の交錯した部分の対応付けには成功していない。また,両眼(多眼)立体視問題を画像の対応付け問題としてではなく,画像中の物体の空間配置問題として,シミュレーテッド・アニーリング法により解く試み²²⁾も行われているが,まだ物体を点で記述する段階にあり,面の記述等が今後の課題とされている。

ここでは,日常生活においてよく見られるこの対応点の順序逆転を含むシーンについても正確な対応付けを可能にする,2段階動的計画法モデルと呼ぶ1つの新しい方法を開発したので,その基本的考え方と行対応問題用の基本アルゴリズムについて記述する。動的計画法を使用した従来手法が対応点のみの探索を行ったのに対して,本方法では対応点の状態を4つに分けて表し,探索を2段階に分けた。即ち,第1段階で順序同一性の仮定の下で対応付く点,左眼でのみ見える点,右眼でのみ見える点を検出し,第2段階で左右走査線上で見える順序の入れ替わる手前の物体の対応点を検出できるようにした。

以下,3.2.2に2段階動的計画法モデルの概要を,3.2.3にその対応点探索アルゴリズムを,3.2.4に実験結果を記述する。

3.2.2 2段階動的計画法モデルの概要

本モデルでは,左右走査線内における画素レベルの対応付けを行う。画素レベルの対応付けは,これまで主として計算時間がかかるという理由から敬遠されてきたが,カメラから得られる高階調のカラー(RGB)データをそのまま利用でき,微妙な色合いの差を左右画像の対応付けに積極的に活用できる利点をもつ。また,画面のほぼ全領域にわたって距離情報を得ることができる利点もある。計算時間は,計算機の進歩,並列処理,ハードウェア化²³⁾等によって解決されることを期待し,この章では実際にかかった処理時間を示すにとどめる。

対応付けは,大局的な評価を可能にする動的計画法を用いて行う。ただし,動的計画法をこれまでのように対応付けの観点でのみ使用すると,手法の原理的な制約から,“対応点の順序は入れ替わらない”という仮定を設ける必要があり,Fig. 3.3に示すような手前に縦細の物体のあるシーンでは左右画像の正確な対応付けは不可能になる。これをPhoto. 3.1の例で説明する。Photo. 3.1は,両眼視の走査線内対応問題用として作成したテスト画像で,左右画像の対応付けにはそれぞれの中央の1行が使用される。Photo. 3.1の中央部分の柱の色とその対応関係(正解)をFig. 3.4に示す。①,②は,カメラの左眼でのみ見える領域,⑤,⑥は,右眼でのみ見える領域である。④は,手前の柱(灰色)で,③の後方の柱群と左右画像で見える順序が逆転する。これをFig. 3.5のように縦軸,横軸にそれぞれ左走査線,右走査線上の画素の色を左から順に並べると,動的計画法では,Fig. 3.5の2次元探索平面上で左右画素の色の差(絶対値)の総和を最小にする,正しい対応

(●)をつなぐパスを求める問題として解くことができる。しかし、動的計画法では、最適解を保証する条件であるマルコフ性を満たす必要があるため、パスの探索は、左下から右上に単調上昇する範囲でしか行えない。このため、Fig. 3.5の(1)の部分は(2)の右下への後戻りが必要となり正しく対応付けることができない。この結果、従来の対応点のみを探索する方法では(3)、(4)の部分の誤対応(○)を生む。

そこで、本モデルでは、Fig. 3.6に示すように、左画素*i*と右画素*j*の交点をノード(*i*, *j*)と呼び、各ノードを以下の4つの状態に分けて表し、各状態間の原子パスをFig. 3.6のように定義することによって、対応点の順序逆転を含むシーンについても動的計画法を2回適用すれば、正しい対応を求めることができるようにした。

D_{ij} : *i*と*j*は順序同一性の下で対応が付く。

V_{ij} : *i*は左眼でのみ見える。

H_{ij} : *j*は右眼でのみ見える。

C_{ij} : *i*と*j*は順次同一性の下では対応がはっきりしない。順序同一性の下では順序逆転のために対応付けができない場合と1画素中で物体の色が大きく変化するために相手の画素とうまく対応付けができない場合が該当する。

後者は両隣の色から簡単に検出できるので、以下の説明では省略する。

本モデルでは、第1段階において、3.2.3(1)に示す動的計画法アルゴリズムにより、順序同一性の仮定の下で左下端から右上端への最適パスを得る。このとき、順序同一性の下で対応が付く領域、左眼でのみ見える領域、右眼でのみ見える領域を検出するとともに順序逆転のために対応付けができない領域を順序同一性の下で状態Cとして検出することにより動的計画法のマルコフ性を満足させている。第1段階の最適パスが通過したノードは、それぞれD, V, H, Cのいずれか1つの状態で表され、それがそのノードの対応関係を示している。つまり、Dを通過したならば、*i*と*j*は対応が付いたことを示し、V又はHを通過したならば、それぞれ*i*は左眼でのみ見える領域、*j*は右眼でのみ見える領域であることを示している。Cを通過した場合は、*i*または*j*のどちらかが順序逆転のために順序同一性の下では対応付けできない領域であることを示している。このため、第2段階では、第1段階でCを通過した左右画素、即ちFig. 3.7の網目領域のみを対象として3.2.3(2)に示す動的計画法アルゴリズムを適用することにより最適パスを求める。そして、第1段階で求めた最適パス(Cの部分を除く)と第2段階で求めた最適パス(Fig. 3.7の網目領域のみ)を合成することにより、最終的な最適パスを得る。

3.2.3 対応点探索アルゴリズム

(1) 第1段階の探索

次のように記号を定義する。

N: 左右1走査線上の画素数。

i: 左走査線上の画素番号 (左端=1 ≤ *i* ≤ N)で、*i*=0を*i*=1の左の視野外を表すことにする。

j: 右走査線上の画素番号 (左端=1 ≤ *j* ≤ N)で、*j*=0を*j*=1の左の視野外、*j*=N+1を*j*=Nの

右の視野外を表すことにする。

$m = (i, j)$: 右画素 i と左画素 j で探索平面上に形成されるノードの指標。

$S = (0, 0)$: 左右走査線の左視野外同士の対応を示すノード, パスの始端。

$E = (N, N+1)$: 左走査線の右端と右走査線の右視野外の対応を示すノード, パスの終端。

$g(m, k)$: ノード k からノード m に至る原子パスのコスト。ノード m の D に至る場合は式 (3.3) の $d_{i,j}$, C , H , V に至る場合は, それぞれペナルティコスト P_c , P_H , P_V が使用される。

$G(m)$: ノード S からノード m に至る最適パスのコスト。ノード m の D , C , H , V に至る場合はそれぞれ $D_{i,j}$, $C_{i,j}$, $H_{i,j}$, $V_{i,j}$ が使用される。

左右走査線の最適な対応は, コスト最小の $G(E)$ を与えるパスを求めることにより得られ, それは次の漸化式で計算される。

$$G(m) = \text{Min}_{\{k\}} \{G(m-k) + g(m, m-k)\}$$

$$G(S) = 0 \tag{3.1}$$

この式をノード m の D , C , H , V に至る場合についてそれぞれ記述すると次のようになる。

$$\begin{aligned} & \cdot D_{i,j} = d_{i,j} + \text{Min}_{\left\{ \begin{array}{l} D_{i,j-1} \\ D_{i-1,j-1} \\ D_{i-1,j} \\ C_{i-1,j-1} \\ H_{i-1,j-1} \\ V_{i-1,j-1} \end{array} \right\}}, RD_{i,j} = 1 \\ & \quad (1 \leq i, j \leq N) \\ & \cdot C_{i,j} = P_c + \text{Min}_{\left\{ \begin{array}{l} C_{i-1,j-1} \\ D_{i-1,j-1} \\ H_{i-1,j-1} \\ V_{i-1,j-1} \end{array} \right\}}, RC_{i,j} = 1 \\ & \quad (1 \leq i, j \leq N-1) \\ & \cdot H_{i,j} = P_H + \text{Min}_{\left\{ \begin{array}{l} H_{i,j-1} \\ D_{i,j-1} \\ C_{i-1,j-1} \\ V_{i-1,j-1} \end{array} \right\}}, RH_{i,j} = 1 \\ & \quad (1 \leq i \leq N, \\ & \quad \quad 1 \leq j \leq N+1) \\ & \cdot V_{i,j} = P_V + \text{Min}_{\left\{ \begin{array}{l} V_{i-1,j} \\ D_{i-1,j} \\ C_{i-1,j-1} \\ H_{i-1,j-1} \end{array} \right\}}, RV_{i,j} = 1 \\ & \quad (0 \leq i, j \leq N-1) \end{aligned} \tag{3.2}$$

式 (3.2) の $D_{i,j}$, $C_{i,j}$, $H_{i,j}$, $V_{i,j}$ は, それぞれ Fig. 3.6 のノード $m = (i, j)$ の D , C , H , V にノード S から至るパスのうち, コスト最小のパスのコストを示している。Fig. 3.6 は, D , C , H ,

Vに至る直前の原子パスを示している。ノード $m=(i, j)$ の D, C, H, Vにどの原子パスを経由してきたかは、式(3.2)の $RD_{i,j}$, $RC_{i,j}$, $RH_{i,j}$, $RV_{i,j}$ に記憶される。式(3.2)では記述を簡単にするため、ノード Sを $V_{0,0}$ 、ノード Eを $H_{N,N+1}$ としている。原子パスのコストはそれぞれ次の意味をもつ。 $d_{i,j}$ は、左画素 i と右画素 j が対応付く場合のコストで、式(3.3)に示す左右画素の色の差である。

$$d_{i,j} = \frac{|R_i - R_j| + |G_i - G_j| + |B_i - B_j|}{255 * 3} \quad (3.3)$$

ここで、 (R_i, G_i, B_i) は左画素 i の光の3原色データ、 (R_j, G_j, B_j) は右画素 j の光の3原色データで、それぞれ0から255の値を使用する。ペナルティコスト P_c, P_H, P_V は、それぞれ i と j は順序同一性の下では対応付けできないとする場合のコスト、 i は左眼でのみ見える領域とする場合のコスト、 j は右眼でのみ見える領域とする場合のコストであり、 $P_H = P_V > P_c$ で、対応付く場合の $d_{i,j}$ より大きい値を使用する。

画像を対応付ける最適パスの探索は、次のようにして行う。

- ① 式(3.2)により $D_{i,j}, C_{i,j}, H_{i,j}, V_{i,j}$ を i の小さい番号から、各 i については j の小さい番号から順に求める。このとき、カメラの向きが平行のときは左画素 i と右画素 j が対応する場合、常に $i \geq j$ (ただし、終端ノード $E = H_{N,N+1}$ を除く)であることを利用する。同時に、それぞれに至るコスト最小の原子パスを $RD_{i,j}, RC_{i,j}, RH_{i,j}, RV_{i,j}$ に記憶する。
- ② 終端ノード $E = H_{N,N+1}$ に到達した最適パスをTable 3.1に使用して逆探索する。即ち、Table 3.1で $RH_{N,N+1}$ から $END2$ (ノード S) までのパスを見つける。

第1段階の解は、ここで得られた最適パスに含まれるノード(左画素 i , 右画素 j)と各ノードがD, C, H, Vのいずれかの状態にあるかである。Dの場合は、対応が付いているのでその視差($i-j$)からそのノード(左画素 i と右画素 j)の距離が得られる。H, Vの場合は、それぞれ左画素 i は左眼のみ、右画素 j は右眼のみで見える領域と解釈される。Cの場合は、順序同一性の下では対応付けできない領域である可能性があるため、次の第2段階の探索を行う。

(2) 第2段階の探索

第1段階でCの状態が検出された左画素と右画素のすべての組み合わせ領域をFig. 3.7の網目領域のように定義する。第2段階では、従来の対応点のみを探索する磯道の探索アルゴリズム¹¹⁾をそのまま使用する。ただし、網目領域以外の領域のコストは0とし、網目領域のコストはペナルティコストを P_c から左右画素の色合いの差のコスト $d_{i,j}$ を差し引いた値を使用する。これにより網目領域において、左右画素の色合いの差のコスト $d_{i,j}$ の値がペナルティコスト P_c より小さいノードのみが検出される。これが第2段階の解となる。厳密には網目領域を横切るパスは複数ノードを通過する場合が多いので複数ノードの平均コストが P_c より小さい一連のノードが検出される。この中でコスト $d_{i,j}$ の値が P_c より大きいノードは、第2段階でも対応関係がはっきりしない領域とされる。この結果、順序同一性の下で対応付けできなく、かつ色合いの差のコストがペナルティコスト P_c より最も小さいノード群が順序逆転の可能性のある網目領域において検出され

る。

最終的には、Fig. 3. 7に示したように第1段階で得られた最適パスのうち状態Cを除くパスと第2段階で網目領域中に得られた最適パスを組合せたものが対応結果となる。

3.2.4 実験結果

(1) 行対応問題への適用

3.2.3で示したアルゴリズムをPhoto. 3.1の左右画像（各々の中央の1行）について、対応付けた（行対応問題）結果をFig. 3.8に示す。Photo. 3.1のシーンは、左右走査線上の対応点に順序逆転が生じるケースである。ペナルティコストは、 $P_c=0.08$, $P_v=P_H=0.10$ を使用した。適用にあたって、次の2つの工夫を施した。

- i. 左右端での対応付けを正確にするため、Fig. 3.6の左端を上がるパスの P_v と上端を右に移るパス P_H の値をそれぞれ始端と終端から64画素までともに0.01と小さくした。
- ii. 左画像の連続した m 画素と右画像の連続した n 画素が同一色で対応するとき、それらの対応付けを均等にするため、式(3.2)の $RD_{i,j} = 1, 3$ で記憶するパスに関しては、色合いの差のコスト $d_{i,j}$ にペナルティコスト $P_b=0.03$ をさらに加えた。

Fig. 3.8はカメラの左眼から左走査線上の各画素に写っている物体までの距離（左眼から前方垂直距離）がプロットされている。(a)が第1段階の結果で、従来手法と同じように順序逆転の生ずる手前の柱は認識されていない。(b)が第2段階の結果で、手前の柱がほぼ認識されている。(c)が正解である。(b)において、左手前に実際には存在しない物体（丸枠部分）が誤認識されている。これは、Photo. 3.1の左側のように手前に複数の柱がある場合、それらの柱の後方にはFig. 3.4の②、⑤のように片眼でしか見えない領域が生じるが、それらが偶然にほとんど同じ色を持つとき、3.2.3(2)で示した第2段階のアルゴリズムではそれらに対応付けてしまうからである。この誤認識は、手前の柱によって片方の眼で見えなくなった領域は他方の眼のいかなる領域とも対応しないことによって一般には除去可能である。しかし、その場所に実際に物体がある場合との区別がまだ完全にできていないので、これを除去するアルゴリズムはここでは入れていない。

(2) 画面对応問題への適用

次に、この手法をPhoto. 3.2の左右画像の対応問題に適用した結果をFig. 3.9(a)に示す。Fig. 3.9(a)は、左画像について左カメラの各画素に写っている物体との距離（光軸と垂直な距離）である。1画面（384行×512列）を、単純に384個の行対応問題に分割し、各々1行毎の対応付けによって得られた距離を3次元的に表示した。Fig. 3.9(b)は、上下（左上、左）画像の対応問題に適用した結果で、Fig. 3.9(a)と同じく左画像についての距離を表示した。

ともに、順序逆転の原因となる、手前の宙に浮いた菱形のパイプと右側のX字形の格子が検出されている。ただし、天井や床の格子模様が実際より少し浮き出て見えたり、まだきちんと対応付けができていない部分もある。また3眼視で問題となる左右画像の結果と上下画像の結果が食い違うときの処理については、結果の異なった画素についてその周囲の画素を含めた色の相関を取り、その大きい方を採用してみた。しかし、このシーンはもともと相関法等の従来手法では対

応付けのうまくいかない代表例として使用したもので、当然よい結果は得られなかった。これらは、今後の課題として残される。

(3) 計算時間

この2段階動的計画法モデルによる走査線内探索問題の計算時間は、大型計算機 (VP2600のスカラ計算) で、約 0.4秒である。1画面 (384行*512画素) で、前処理 (データ読み込み) と後処理 (距離計算, 各種出力) を含めて約 180秒である。視差の最大値を仮定し、探索範囲を限定すれば、2-5倍の高速化は可能である。

3.2.5 おわりに

この項では、これまで左右画像の対応付けが成功していない順序逆転を含むシーンについて、対応付けを可能にする新しい方法を提案し、走査線内対応問題、画面对応問題でその有効性を示した。この方法は、画素対応法のみでなく、これまで研究の進んでいる区間対応法等にも応用可能と考えている。

ここでは、新しい対応付けのロジックを走査線内対応問題で示しただけであり、多眼視対応問題への拡張、自然画像へ適用した場合のペナルティ・コストの適正值評価等が今後の課題として残されている。

両眼立体視を実用化するためには、ここで使用した画素 (色) レベルの情報のみでなく、輪郭線 (エッジ) レベルの情報や物体に関する知識等を高速に融合する手法が最終的には必要と考えられるが、これらも今後の課題である。

3.3 工具認識用画像データベースの試作

HASPでは、半既知環境下の物体認識方法をFig. 3.2に示すように考えている。シーン中の物体は、両眼立体視により得られた3次元空間情報 (距離画像: 各画素の値は、カメラからそこに写っている物体までの距離を示す。) と色情報 (色画像: 各画素の値は、そこに写っている物体のRGB値を示す。) を利用して、他の物体と分離・抽出される。ロボットは、抽出された物体情報からその物体が何かを、物体の輪郭や特徴をキーとして視覚認識用知識ベースを利用して探る。このとき、物体の一部は、他の物体に隠れていることがあり、抽出された物体からは欠けていることがある。また、物体は、見る方向によってその形状等が異なる。

平成3年度は、工具を対象とした視覚認識用画像データベースを試作した。Table 3.2に示す15個の工具と13種類の文具 (文具には、色違いがある。) について各々26方向 (45° 毎) から見た画像データベースを作成した。一例をPhoto. 3.3に示す。Photo. 3.3は、ペンチを正面から垂直方向に45° ずつ視点を変えて見たときの画像である。左右それぞれ9方向から見た画像と合わせて1物体26画像を作成した。画像データは、実物の寸法を測定し、3次元物体モデリング用ソフト: FUSION¹⁾ を使用してソリッドモデルで記述し、レイ・トレーシング・アルゴリズムによって作成している。この画像データベースを使用して、最終的には、例えばPhoto. 3.4に示す画像からペンチ

(a)や赤いマジック(b)を認識させたい。

基本的には、抽出した物体画像に対して、大きさと画像中の位置に影響を受けないような変換をかけ、それと同じ変換をかけた画像データベースを探索し、一番似た特性をもつ画像データが得られればよい。このとき、変換領域でのデータ量は、原画像のデータ量と比べて大幅に削減させる必要がある。画像変換手法の研究と各物体の特徴抽出が今後の課題となる。

3.4 今後の研究方向

HASPにおける視覚認識の研究は、環境に関する知識を利用したモデル・ベースドな研究と実際のハードウェア（CCDカメラ等）を利用した実世界を対象とする研究の両方を今後行っていく必要がある。

平成4年度は、3次元空間認識の研究を継続するとともに、原子力施設用の巡回点検ロボットに必要な視覚センサ及びその情報処理システムの開発を行う予定である。

参 考 文 献

- 1) 浅井 他：原子力知能化システム技術の研究（人間動作シミュレーション・プログラム：HASP）－昭和63年度報告書－，日本原子力研究所，JAERI-M 89-023(1989. 3)。
- 2) 浅井 他：原子力知能化システム技術の研究（人間動作シミュレーション・プログラム：HASP）－平成元年度報告書－，日本原子力研究所，JAERI-M 90-060(1990. 3)。
- 3) 浅井 他：原子力知能化システム技術の研究（人間動作シミュレーション・プログラム：HASP）－平成2年度報告書－，日本原子力研究所，JAERI-M 91-101(1991. 6)。
- 4) Asai, K. et al.:HASP:Human Acts Simulation Program, Proc. of the 1st International Conf. on Supercomputing in Nuclear Appli. (SNA'90), pp.611-615, NEDAC(1990. 3)。
- 5) Asai, K. et al.:A Study on Intelligent Robots in Nuclear Systems, ICHMT 2nd International Forum, pp.9L1-9L8(1992. 3)。
- 6) 日本ロボット学会誌[特集] ロボットビジョン, Vol.10, No.2, (1992. 4)。
- 7) 藤井：ニューロ手法と画像認識, 富士通, サイエントフィック・システム研究会 News-letter, No.50, pp.7-62(1990)。
- 8) Hopfield, J.J. and Tank, D.W.:Neural computation of decisions in optimization problems, Biol. Cybern., Vol.52, pp.141-152(1985)。
- 9) 谷内田, 石黒：ロボットの視覚, 人工知能, Vol.5, No.6, pp.720-730(Nov. 1990)。
- 10) 大田, 山田：動的計画法によるパターンマッチング, 情報処理, Vol.30, No.9, pp.1058-1066 (Sep. 1989)。
- 11) 磯道, 竹政：航空写真からの高度図作製, 第7回画像工学コンファレンス, pp.31-34(1976)。
- 12) 佐々木, 古川：エッジに着目した動的計画法によるステレオ画像の対応点検出, 信学技法, IE78-97, pp.47-56(1979)。
- 13) 大田, 正井, 池田：動的計画法によるステレオ画像の区間対応法, 信学論, Vol. J68-D, No.4, pp.554-561(1985. 4)。
- 14) 大田, 金出：走査線間の整合性を考慮した2段の動的計画法によるステレオ対応探索, 情報処理, Vol.26, No.11, pp.1356-1363(Nov. 1985)。
- 15) Ohta, Y. and Kanade, T.:Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming, IEEE Trans., Vol. PAMI-7, No. 2, pp.139-154(March 1985)。
- 16) 奥富, 金出：統計的モデルに基づく適応形ウィンドウによるステレオマッチング, 信学論, Vol. J74-D-II, No.6, pp.669-677(1991. 6)。
- 17) Marr, D. and T. Poggio: Cooperative Computation of Stereo Disparity, Science, Vol.194, pp.283-287(Oct. 1976)。

- 18) 佐藤, 粟屋, 田村: 両眼立体視のエネルギー最小化アプローチ, 信学論, Vol. J71-D, No. 2, pp. 315-326(1988. 2)。
- 19) 西元, 白井: 多重分解能での, 差のヒストグラムを用いた特徴ベース法の両眼立体視モデル, ロボット誌, Vol. 4, No. 6, pp. 607-617(1986. 12)。
- 20) 北村, 谷内田: 三眼視による三次元情報の計測, ロボット誌, Vol. 5, No. 2, pp. 131-138 (1987. 4)。
- 21) Dhond, U.R. and Aggarwal, J.K.: Structure from Stereo-A Review, IEEE Trans. Sys. Man and Cyber., Vol. 19, No. 6, pp. 1489-1510(Nov. 1989)。
- 22) 伊藤, 藤井: 知能ロボットにおけるセンサー融合による状況認識, 第7回「原子力におけるソフトウェア開発」研究会報告集 (JAERI-M 92-054), pp. 76-85(Mar. 1992)。
- 23) 大田, 高野, 池田: ステレオ画像における区間対応探索法の高速度ハードウェア処理, 情報処理論文誌, Vol. 28, No. 2, pp. 168-176(Feb. 1987)。

Table 3.1 Selection method of the optimal path.

変数	値	最適パス	処理	次処理	例外処理
RD _{i,j}	1	D _{i,j-1}	j=j-1	RD _{i,j}	j ≤ 1ならRD _{i,1} の処理
	2	D _{i-1,j-1}	i=i-1, j=j-1	RD _{i,j}	j ≤ 1ならRD _{i,1} の処理
	3	D _{i-1,j}	i=i-1	RD _{i,j}	i < jならエラー
	4	C _{i-1,j-1}	i=i-1, j=j-1	RC _{i,j}	j ≤ 1ならRC _{i,1} の処理
	5	H _{i-1,j-1}	i=i-1, j=j-1	RH _{i,j}	j ≤ 1ならRH _{i,1} の処理
	6	V _{i-1,j-1}	i=i-1, j=j-1	RV _{i,j}	j ≤ 1ならRV _{i,1} の処理
RC _{i,j}	1	C _{i-1,j-1}	i=i-1, j=j-1	RC _{i,j}	j ≤ 1ならRC _{i,1} の処理
	2	D _{i-1,j-1}	i=i-1, j=j-1	RD _{i,j}	j ≤ 1ならRD _{i,1} の処理
	3	H _{i-1,j-1}	i=i-1, j=j-1	RH _{i,j}	j ≤ 1ならRH _{i,1} の処理
	4	V _{i-1,j-1}	i=i-1, j=j-1	RV _{i,j}	j ≤ 1ならRV _{i,1} の処理
RH _{i,j}	1	H _{i,j-1}	j=j-1	RH _{i,j}	j ≤ 1ならRH _{i,1} の処理
	2	D _{i,j-1}	j=j-1	RD _{i,j}	j ≤ 1ならRD _{i,1} の処理
	3	C _{i-1,j-1}	i=i-1, j=j-1	RC _{i,j}	j ≤ 1ならRC _{i,1} の処理
	4	V _{i-1,j-1}	i=i-1, j=j-1	RV _{i,j}	j ≤ 1ならRV _{i,1} の処理
RV _{i,j}	1	V _{i-1,j}	i=i-1	RV _{i,j}	i < jならエラー
	2	D _{i-1,j}	i=i-1	RD _{i,j}	j ≤ 1ならRD _{i,1} の処理
	3	C _{i-1,j-1}	i=i-1, j=j-1	RC _{i,j}	j ≤ 1ならRC _{i,1} の処理
	4	H _{i-1,j-1}	i=i-1, j=j-1	RH _{i,j}	j ≤ 1ならRH _{i,1} の処理
RD _{i,1}	1	D _{i-1,1}	i=i-1	RD _{i,1}	i ≤ 1ならEND2の処理
	2	V _{i-1,0}	i=i-1, j=0	END1	
RC _{i,1}		V _{i-1,0}	i=i-1, j=0	END1	
RH _{i,1}		V _{i-1,0}	i=i-1, j=0	END1	
RV _{i,1}	1	V _{i-1,1}	i=i-1	RV _{i,1}	i ≤ 1ならエラー
	2	D _{i-1,1}	i=i-1	RD _{i,1}	
END1		V _{i-1,0}	i=i-1	END1	i ≤ 1ならEND2の処理
END2				終了	

Table 3.2 List of tools and stationeries.

No	工具物体名	No	文房具物体名	備考
1	ペンチ	1	ボールペン	赤, 黒
2	ラジオペンチ	2	蛍光ペン	赤, 青, 黄
3	ニッパー	3	ボクシーペン	赤, 青, 緑
4	プライヤー	4	マジックインキ (細書き用)	赤, 青
5	プラスドライバー (大)	5	マジックインキ (大型)	赤, 青, 黄
6	プラスドライバー (小)	6	色鉛筆	
7	マイナスドライバー (大)	7	鉛筆	
8	マイナスドライバー (小)	8	ケシゴム	
9	両口スパナ (大)	9	ブリット糊	
10	両口スパナ (中)	10	ハサミ	
11	両口スパナ (小)	11	ホッチキス	
12	メガネレンチ (大)	12	スコッチテープ	
13	メガネレンチ (中)	13	ミスノン	
14	メガネレンチ (小)			
15	金槌 (頭が金属, 柄が木)			

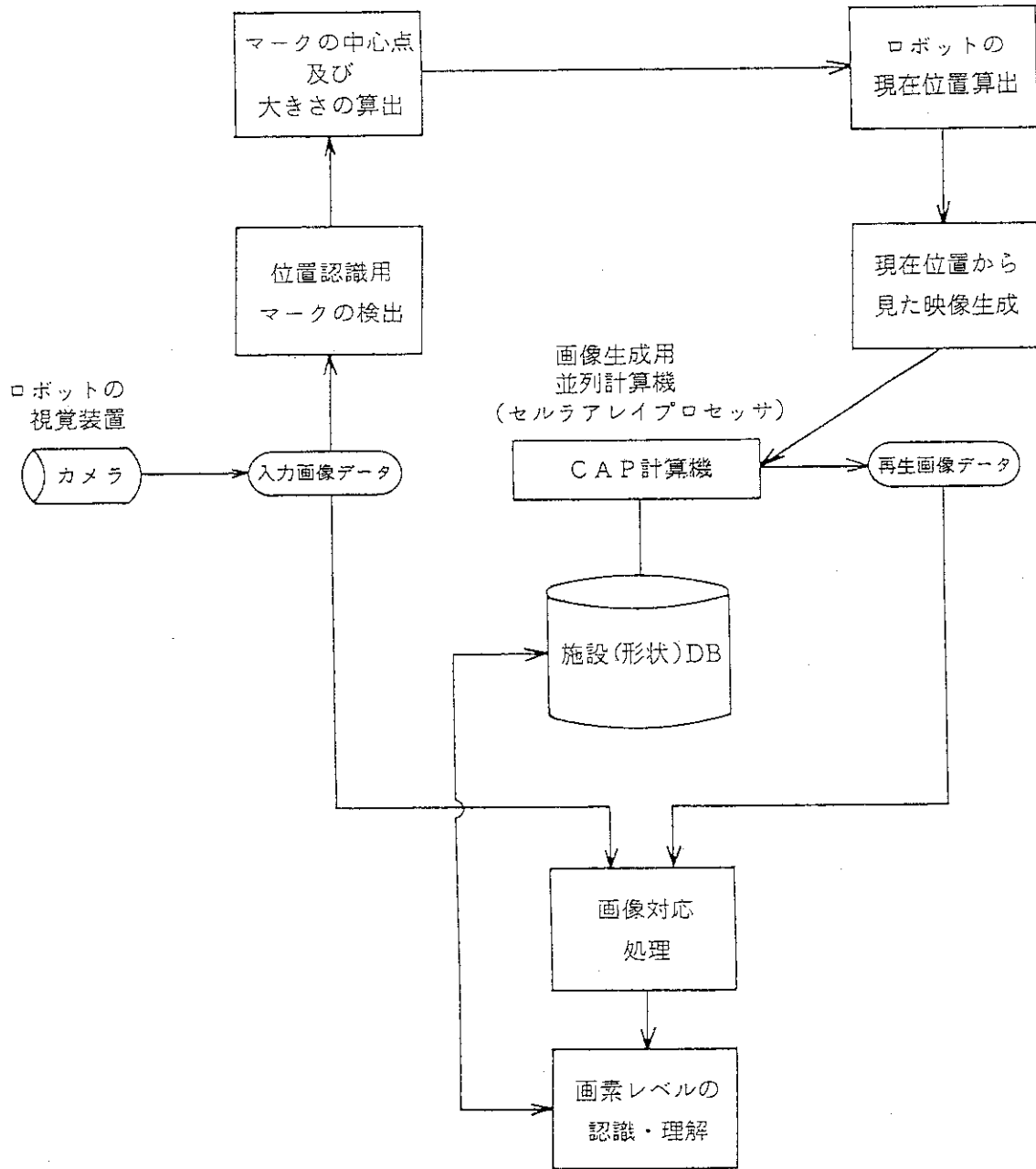


Fig. 3.1 An understanding process of the scene in a structured environment.

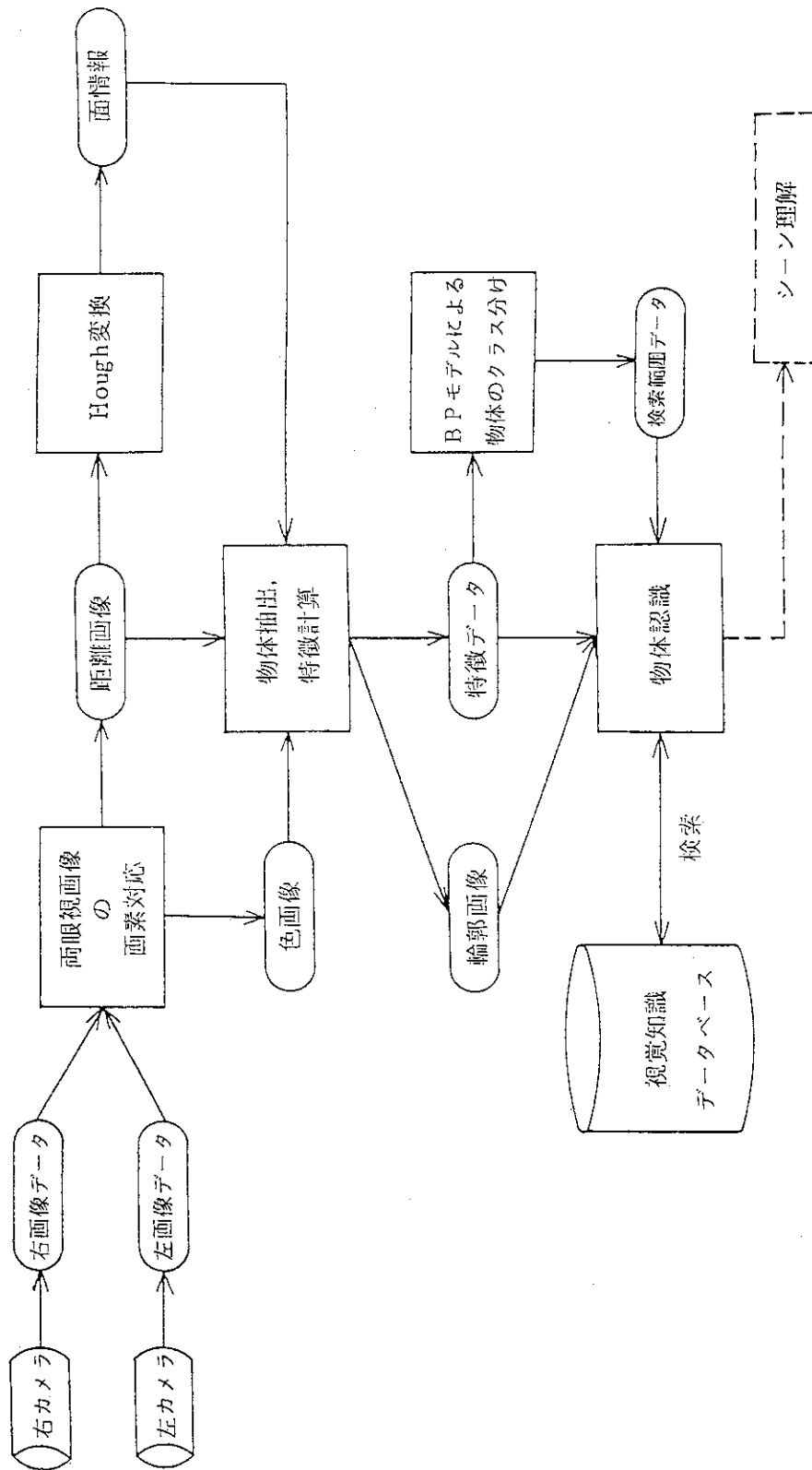


Fig. 3.2 An understanding process of the scene in a semi-structured environment.

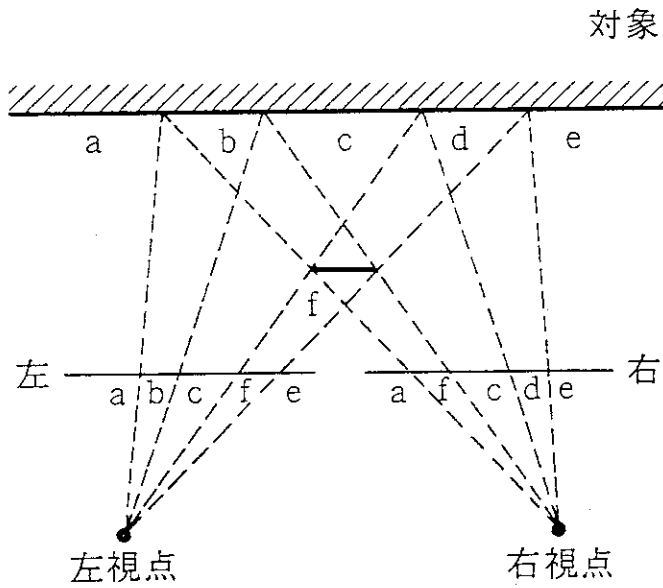


Fig. 3.3 An example of positional reversal of elements between the right and left scanlines¹⁰⁾.

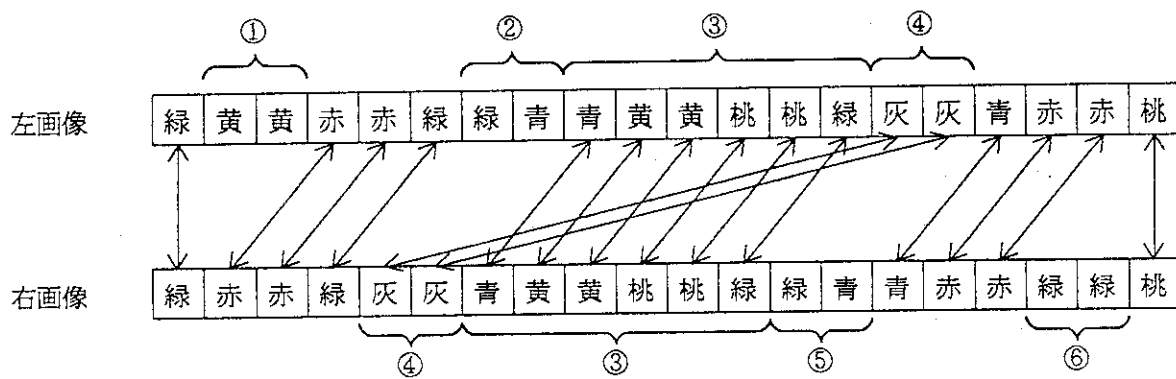


Fig. 3.4 Correspondence of the pixels between the right and left scanlines.

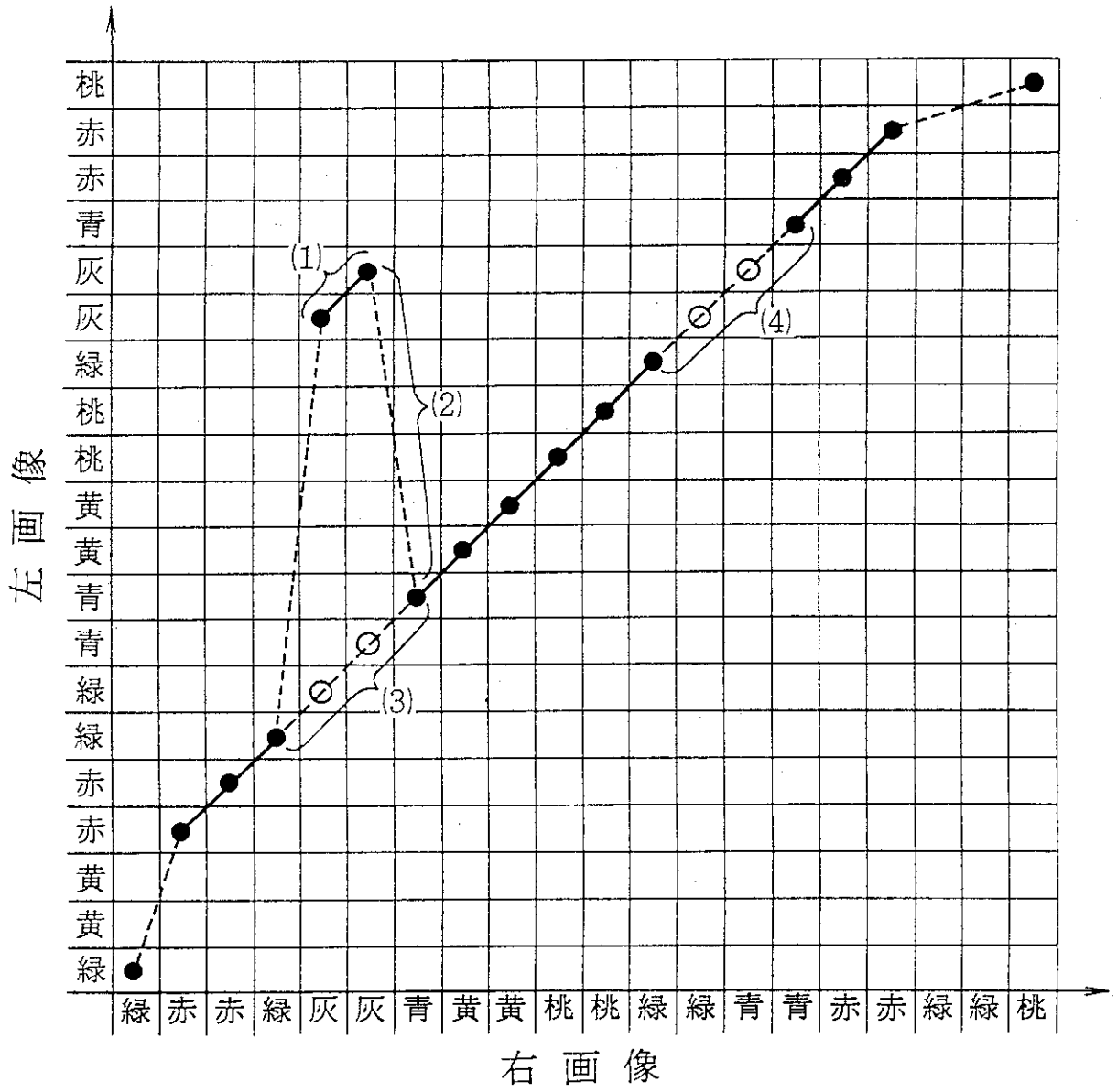


Fig. 3.5 Two-dimensional search plane.

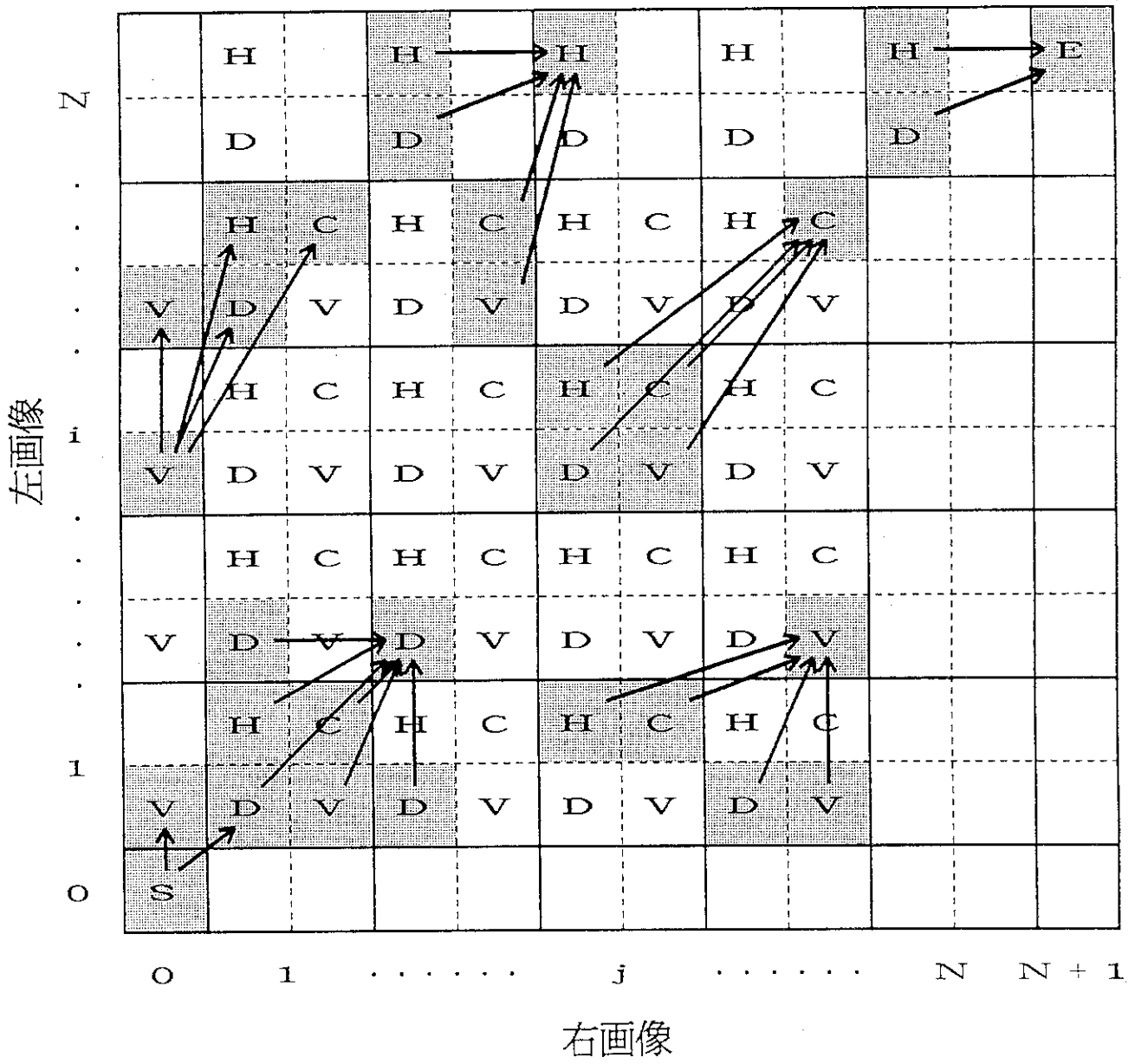


Fig. 3.6 Primitive path.

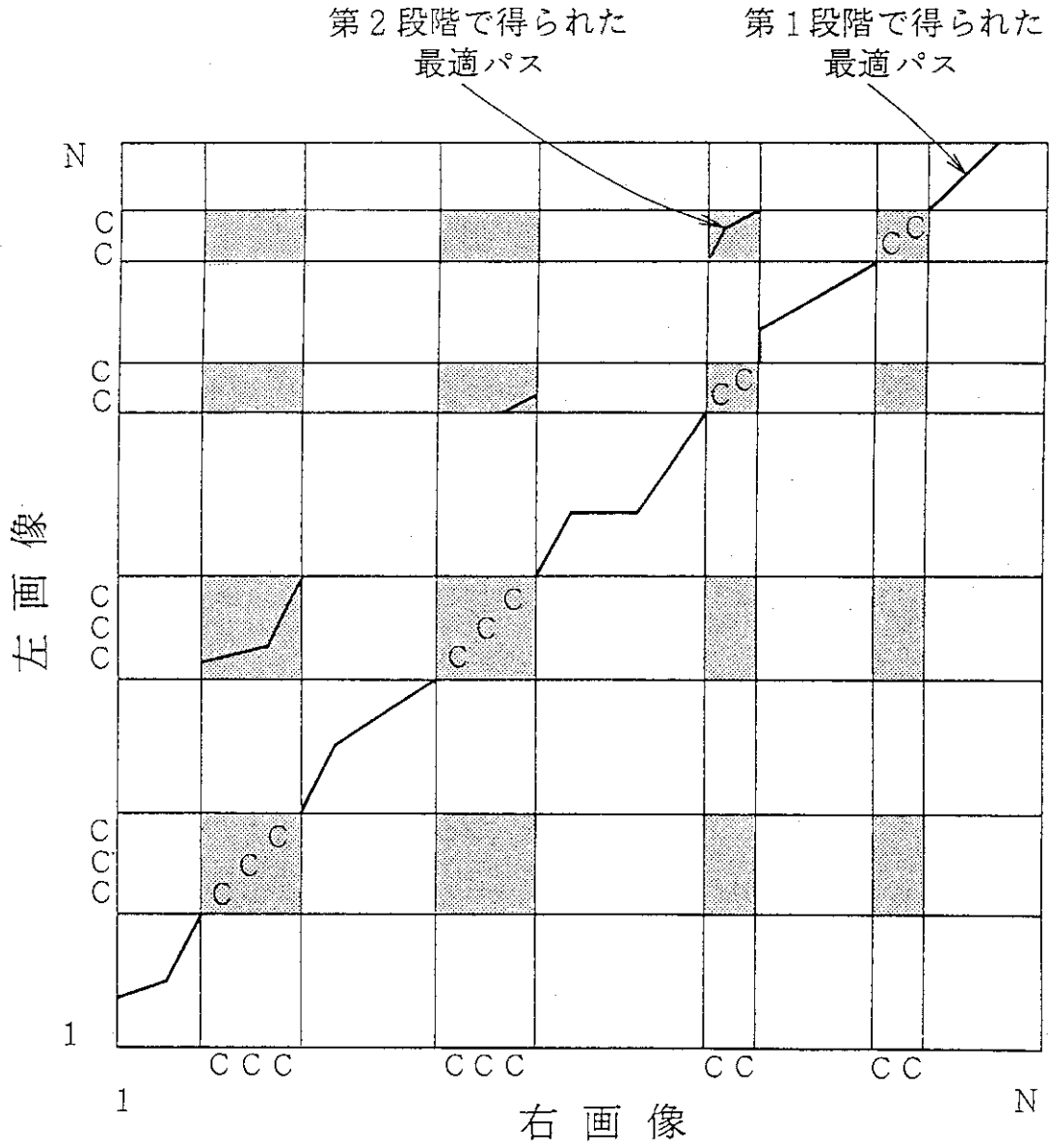
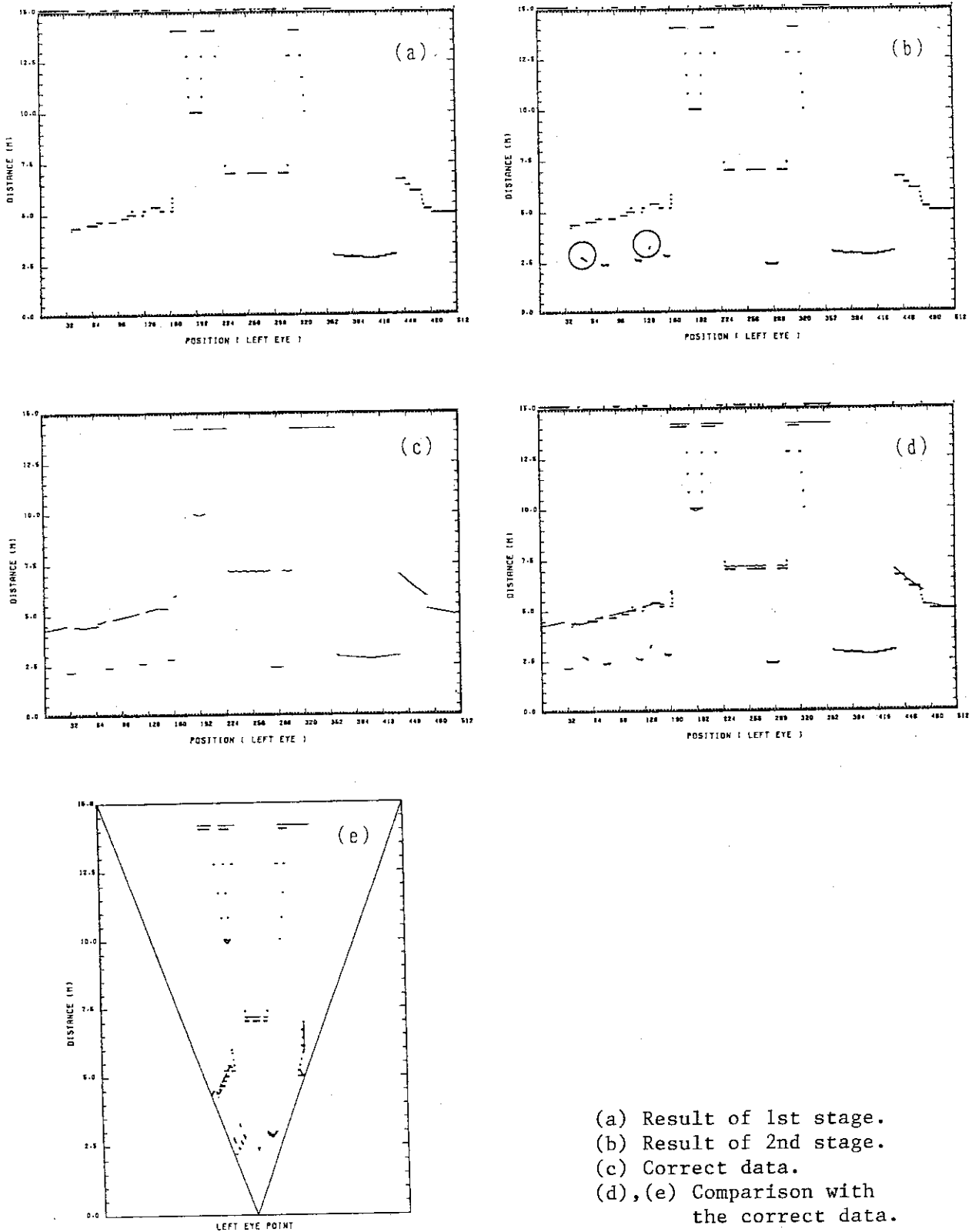
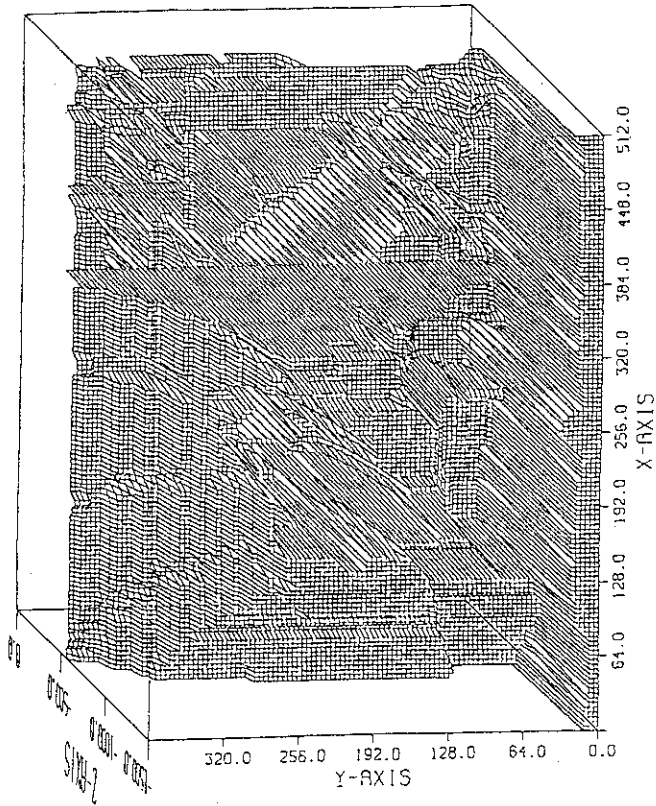


Fig. 3.7 Region for the second search.

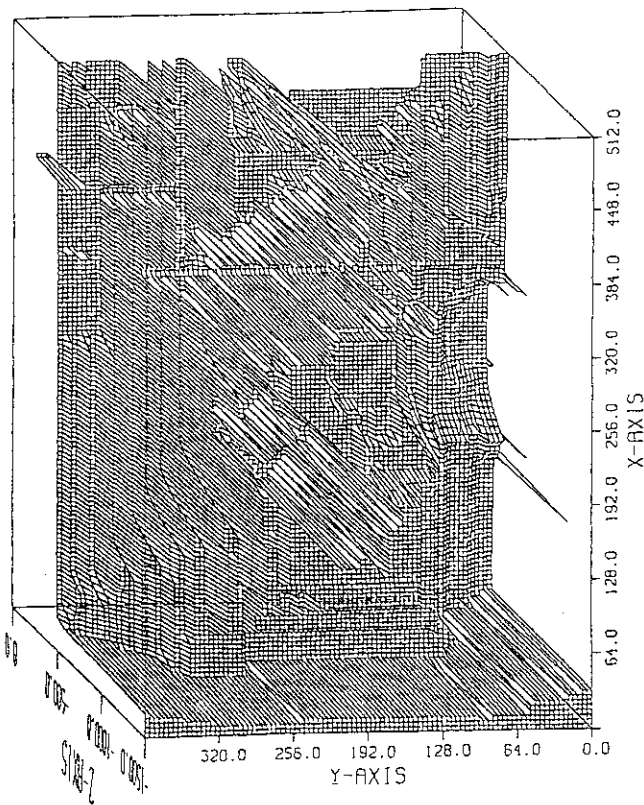


(a) Result of 1st stage.
 (b) Result of 2nd stage.
 (c) Correct data.
 (d),(e) Comparison with
 the correct data.

Fig. 3.8 Distance map of the left scanline by this method and correct distance map.

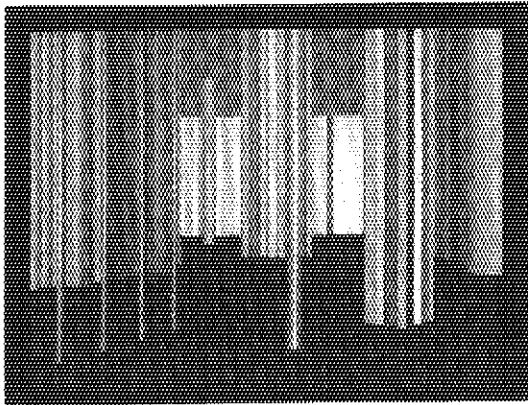


(b)

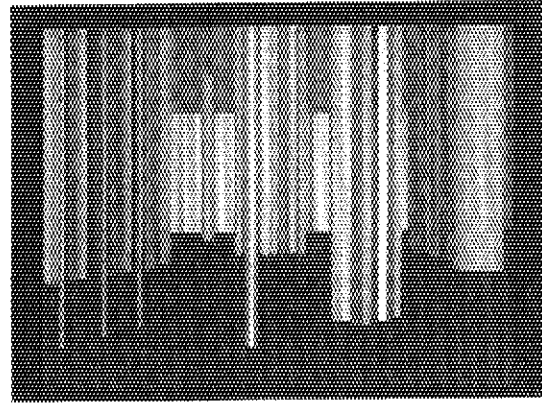


(a)

Fig. 3.9 Distance map of the left image by this method.
(a) Result of the correspondence between right and left images.
(b) Result of the correspondence between left and upper-left images.

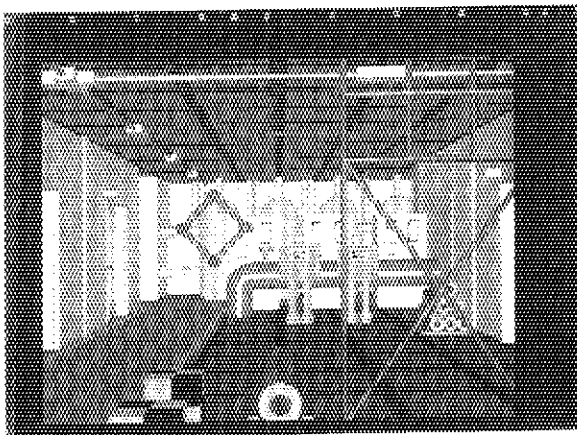


(a) Left image.

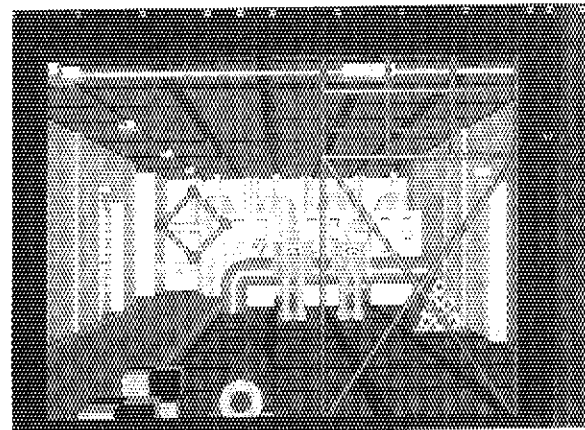


(b) Right image.

Photo. 3.1 A test images of the scanline stereo matching problem.



(a) Left image.



(b) Right image.

Photo. 3.2 A test images of the stereo matching problem.

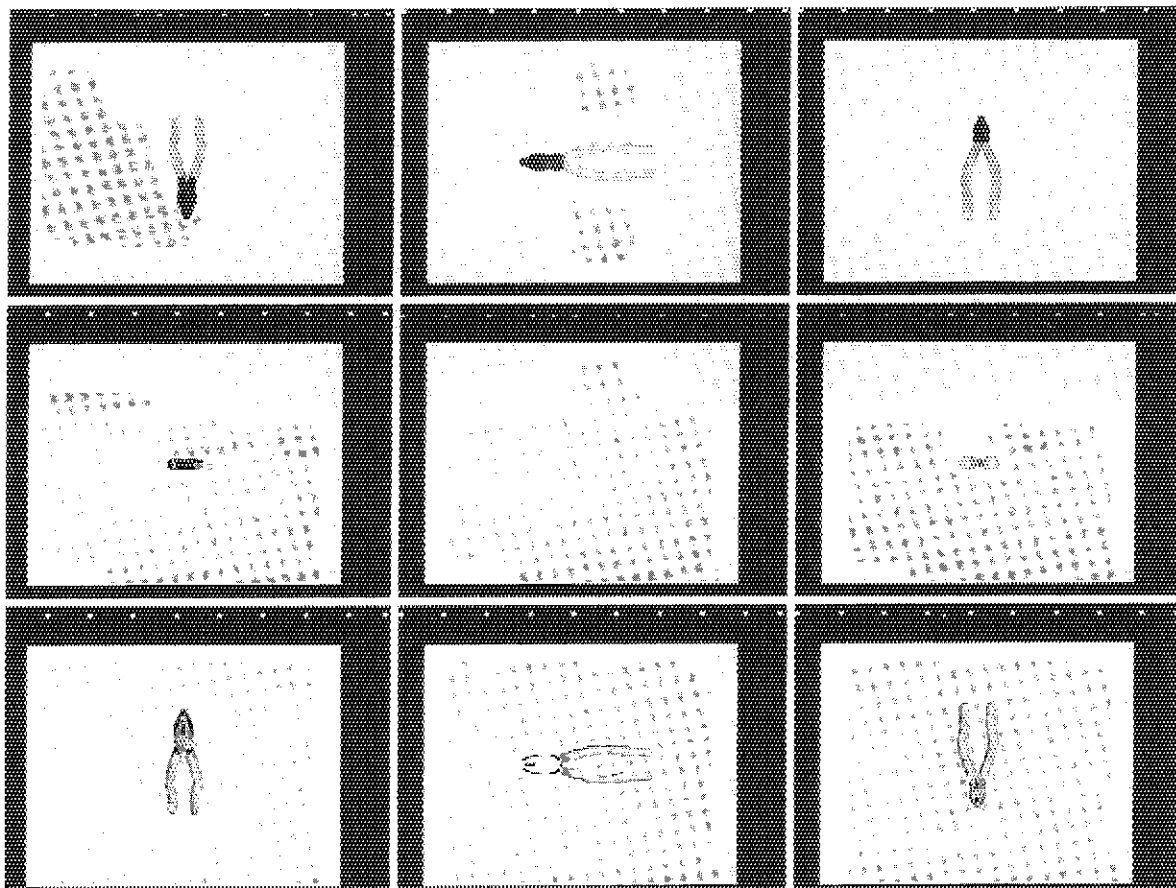


Photo. 3.3 A part of the images for a cutting pliers.

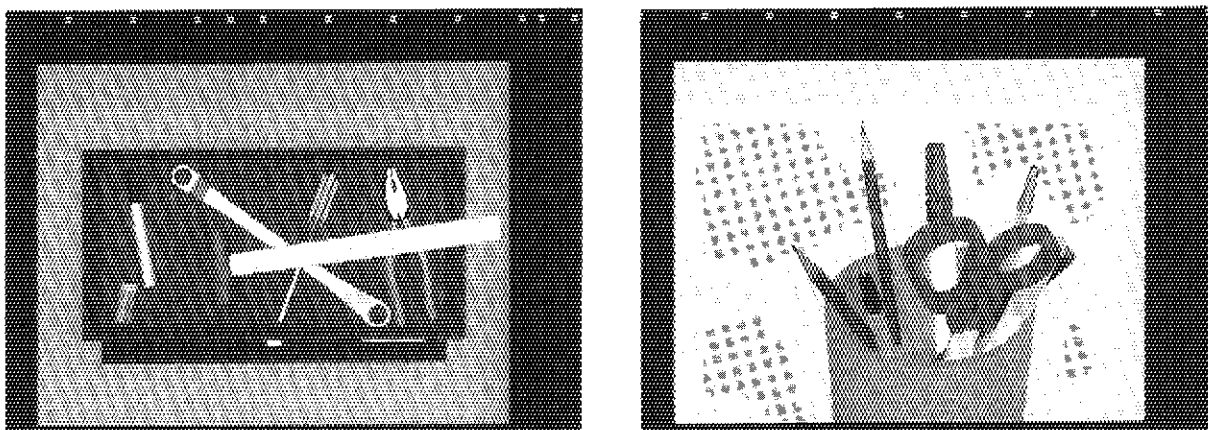


Photo. 3.4 Two examples of the images for the image understanding. (a) An image of tools. (b) An image of stationeries.

4. 人間形ロボットの2足歩行シミュレーション

4.1 はじめに

人間動作シミュレーション技術の研究¹⁾（以下、HASP）では、原子力プラントにおける巡回点検用人間形知能ロボットの研究開発を進めており、巡回点検のためのセンシング・システムを搭載した、2足歩行ロボット・システムの実現を目指している。本章ではこのロボット設計に資する基盤技術開発を目的とした、ロボットの動作シミュレーションについて報告する。

HASPでは、任意位置移動を行う移動機構として2足歩行動作を取り上げている。2足歩行ロボットに関する研究は国内の大学を中心に、いくつかの研究機関で行われているが（Table 4.1）、従来の研究ではどちらかと言うとハードウェアから出発し、与えられたハードウェア的条件の基に歩行安定性を考察してきた。本研究は、ソフトウェアからのアプローチであり、2足歩行をハードウェアで実現する前に、数値シミュレーションによってロボット設計パラメータの考察を行い、歩行安定性を十分に検討する。すなわち、2足歩行ロボット設計に有用なソフトウェアを開発することを目的としている。

2足歩行には、定常歩行、発進・停止歩行、方向転換、階段／傾斜地の歩行などのいろいろな歩行パターンが要求される。ロボット設計上重要となるこれらの安定歩行パターンを、ロボットの機構を考慮し数値シミュレーションによって検討する。平成2年度までは、人間形2足歩行ロボット動作シミュレーション・モデルとして、上肢有りモデル：腕固定、腕可動〔受動、能動〕を開発し、直進定常歩行、階段歩行、発進・停止歩行のシミュレーション及び腕動作の定常歩行動作への影響等について検討を行った^{1)・2)}。また、今年度新たに上肢なしモデルを開発し、歩容と補償動作について、定常歩行を中心とした直進歩行のシミュレーションを行った³⁾。本モデルでは歩容から補償動作（上体動作）、逆に補償動作（上体動作）から歩容（補償歩容）を求めることが可能である。本報では、今年度開発したモデルの力学モデルの概要、その数値解法、ハードウェア化のために行った検討結果について述べる。

4.2 2足歩行ロボット・モデル

4.2.1 ロボット数学モデルの概要

今年度新たに開発した上肢なしモデルをFig. 4.1に示す。このモデルは前年度までに開発した人間形モデル²⁾を基本にしたもので、上体おもり、アクチュエータ等を質点とし、リンク構造部材を剛体とする準質点系モデルである。なお、Fig. 4.1に示した値はシミュレーションのための基準データであり、任意に設定可能である。

一般にロボットをモデル化する場合、構造部材の考慮から剛体系での記述が普通であるが、質量

がボディの一部に集中するような場合（例えば、アクチュエータの質量がリンクのそれに比較して非常に大きいような場合）は質点系での記述も可能である。本モデルは剛体系・質点系のどちらでも記述が可能で、シミュレーション段階ではその大きさや慣性モーメントが不明確であるアクチュエータやセンシング・システム等を質点とし、構造部材を剛体としており、これらの混在するモデルを許している。また、本2足歩行ロボット・モデルは8ボディ、7関節からなり、下肢の前頭面内の動きは両脚（legs）が平行で、足（foot）は常に床面と水平であると仮定し、下肢5自由度、（ピッチ軸4自由度、ロール軸1自由度）、腰2自由度（ピッチ軸・ロール軸それぞれ1自由度）の合計7自由度を持つ（Fig. 4.2）。但し、現段階では直進歩行を前提とし方向転換の自由度は与えていない。

4.2.2 ロボット運動方程式

一般に剛体系の運動方程式は（4.1）で表せる。

$$\mathbf{M} = \sum_{i=1}^n (\mathbf{r}_i - \Delta \mathbf{r}) \times m_i \mathbf{G} + \sum_{i=1}^n (\mathbf{r}_i - \Delta \mathbf{r}) \times m_i \frac{d^2 \mathbf{r}_i}{dt^2} + \sum_{i=1}^n \frac{d(\mathbf{J}_i \cdot \boldsymbol{\omega}_i)}{dt} \quad (4.1)$$

$\Delta \mathbf{r} = (X, Y, Z)$: 任意の点の位置ベクトル,

$\mathbf{M} = (M_X, M_Y, M_Z)$: $\Delta \mathbf{r}$ におけるモーメント,

n : ボディの数,

m_i : ボディ i の質量,

$\mathbf{r}_i = (x_i, y_i, z_i)$: ボディ i の重心位置ベクトル,

$\mathbf{G} = (0, 0, g)$: 重力加速度ベクトル,

$\mathbf{J}_i = \text{diag}(J_{X_i}, J_{Y_i}, J_{Z_i})$: 慣性テンソル,

$\boldsymbol{\omega}_i = (\omega_{X_i}, \omega_{Y_i}, \omega_{Z_i})$: ボディ i の角速度ベクトル

なお、ここではアクチュエータ、おもり等の質点もボディの一つとして数え（ボディ番号についてはFig. 4.2参照）、質点部については慣性テンソルを0行列として定式化を行う。

式(4.1)に関し床面上において、左辺のX成分とY成分が共に零になる点 $\Delta \mathbf{r}$ を、Zero Moment Point(ZMP)と呼び歩行中の安定な状態の規範として用いた。このZMPがロボットの足底接地面の形成する多角形の安定領域内に存在していれば、ロボットは安定に歩行できる⁴⁾。

また、式(4.1)は次の微分方程式の形となる⁵⁾。

$$\{\mathbf{M}\} = [\mathbf{A}]\{\ddot{\boldsymbol{\Xi}}\} + [\mathbf{B}]\{\dot{\boldsymbol{\Xi}}^2\} + [\mathbf{C}]\{\boldsymbol{\Xi}\dot{\boldsymbol{\Xi}}\} + \{\mathbf{G}\} \quad (4.2)$$

ここで、一般化された駆動力ベクトルを $\{\mathbf{M}\}$ 、状態変数（関節角変数）を $\{\boldsymbol{\Xi}\}$ で表し、係数行列 $[\mathbf{A}]$ 、 $[\mathbf{B}]$ 、 $[\mathbf{C}]$ 及びベクトル $\{\mathbf{G}\}$ は、定式化に使われた一般化座標の関数である。右辺第一項は慣性項、第二項は遠心力項、第三項はコリオリ力項、第四項は重力項である。本研究のZMP規範型の場合、前述のように運動方程式に対する拘束条件が、ZMPにおける式(4.1)のX成分

とY成分が0であるという2つであり、一部の運動を規定し、安定な歩行パターンとなるように補償動作を計算することになる。式(4.2)の運動及び駆動力を既知な部分 $\{\Xi_0\}$ 、 $\{M_0\}$ と、未知な部分 $\{\Xi_U\}$ 、 $\{M_U\}$ に分離し、未知量について解くと式(4.3)、式(4.4)のようになる。

$$\begin{aligned} \{\ddot{\Xi}_U\} &= [A_{UU}]^{-1}(\{M_0\} - [A_{00}]\{\ddot{\Xi}_0\} \\ &\quad - [P_0]([B]\{\dot{\Xi}^2\} + [C]\{\dot{\Xi}\dot{\Xi}\} + \{G\})) \end{aligned} \quad (4.3)$$

$$\begin{aligned} \{M_U\} &= [A_{UU}]\{\ddot{\Xi}_U\} + [A_{U0}]\{\ddot{\Xi}_0\} \\ &\quad + [P_U]([B]\{\dot{\Xi}^2\} + [C]\{\dot{\Xi}\dot{\Xi}\} + \{G\}) \end{aligned} \quad (4.4)$$

ここで、各係数行列は以下のような意味を持つ。

$$\begin{bmatrix} P_0 \\ P_U \end{bmatrix} \{M\} = \begin{bmatrix} M_0 \\ M_U \end{bmatrix} \quad (4.5)$$

$$[R_0 \mid R_U] \begin{bmatrix} \Xi_0 \\ \Xi_U \end{bmatrix} = \{\Xi\} \quad (4.6)$$

$$\begin{bmatrix} P_0 \\ P_U \end{bmatrix} \{A\} [R_0 \mid R_U] = \begin{bmatrix} A_{00} & A_{0U} \\ A_{U0} & A_{UU} \end{bmatrix} \quad (4.7)$$

4.2.3 上体補償動作の計算法 (上体補償型モデル)

このモデルは定常歩行における任意の歩容に対する補償動作を上体で行うものである。この計算のフローチャートをFig. 4.3に示す。

(a) 支持足首強度計算

はじめに、ロボット・ボディ・データ：構造部材の材質、寸法及び質量(質点部)を入力し、単脚支持状態での支持足首における強度計算を行う。ロボット・システム全体をFig. 4.4のようにモデル化すると足首部における引張応力は以下のように表せる⁶⁾。

$$\sigma = \frac{32Wa \sin\omega t}{\pi d^3} + \frac{4W \cos\omega t}{\pi d^2} \quad (4.8)$$

式(4.8)で得られた結果と構造部材の引張強さから、足首部アクチュエータ上部のリンク直径を決定する。本研究でのロボット・モデルは、比強さ、比剛性、コスト、加工の容易さなどの考慮から材質をアルミニウム合金と仮定した(強度、軽量化だけを考慮すればCFRP等に代表される強化繊維を用いるのがベターである。また、航空機等に用いられるチタン系の材料は高価であり高温で酸化されやすい)。アルミニウム合金は降伏点が鋼などに比べ低いため、安全率を5程度としてリンク径を決定する。なお、ここでは簡単のためリンクを円柱と仮定しているが、円筒形状とすることにより同じ強度でも軽量化することが可能である。他の部分のリンク径については、強度的に安全となる、最大荷重のかかる足首部アクチュエータ上部でのリンク径と同一とする。

(b) 基本歩行パターンの作成

次に、下肢の歩容及び ZMP軌跡（これらを合わせて基本歩行パターンと呼ぶ）を作成する。昨年度までは、これらのデータについては任意のサンプル点における関節角度データを入力しスプライン関数などで補間して作成していたが、床面との接地時の相対速度の調整や床面との干渉回避がやや施行錯誤的であり困難であること、また両脚支持期を持たないことなどの問題点があった。そこで、今年度新たにこれらの問題点を考慮し、歩容及び ZMP軌跡を自動生成するソフトウェアの開発を行った。このプログラムでは、歩容データ：歩行周期、歩幅、腰の初期位置・速度、両脚支持時間及び遊脚の足の通過点から、遊脚の足の軌跡（支持脚の足は常に床面と全点接地しているものとする）、逆運動学計算による歩容、安定領域内を滑らかに移動する ZMP軌跡を作成することが可能である。

遊脚の足の軌跡の作成には、床面との接地時の境界条件（相対速度、加速度がともに 0）などを考慮しスプライン関数で補間して作成する（スプライン関数の次数は任意に設定可能）。腰の軌跡については、ここでは一定高を等速度（定常歩行時）で動くとして仮定した。これは、搭載する視覚センサ等への振動の影響を考慮したことによるものであるが、最適な歩容について考えるならば、腰の軌跡を何らかの評価関数（具体的な規範はまだ無いが、エネルギー最少など）のもとで最適化する必要がある。また、腰の動作による安定歩行も可能であると考えられる。これらについては、このシステムの改良により容易に行える。但し、評価関数の定義については十分に検討する必要がある。

遊脚の足の軌跡及び腰の軌跡が決まれば幾何学的に膝関節の位置が決定され、逆運動学計算によって下肢の各関節角度を求めることができる。ロボットを Fig. 4.5 に示す矢状面（支持足首を原点とした座標系：足首座標系）で考えれば、各関節角度は以下のように求められる¹⁾。

$$\begin{aligned}
 \beta_{1L} &= \tan^{-1} \left[\frac{x_8 R_2 - z_8 \sqrt{4l_2^2 r_1^2 - R_1^2}}{z_8 R_2 + x_8 \sqrt{4l_2^2 r_1^2 - R_1^2}} \right] \\
 \beta_{2L} &= \tan^{-1} \left[\frac{x_8 R_1 + z_8 \sqrt{4l_2^2 r_1^2 - R_1^2}}{z_8 R_1 - x_8 \sqrt{4l_2^2 r_1^2 - R_1^2}} \right] \\
 \beta_{1R} &= \tan^{-1} \left[\frac{(x_8 - x_{15}) R_4 - (z_8 - z_{15}) \sqrt{4l_2^2 r_2^2 - R_5^2}}{(z_8 - z_{15}) R_4 + (x_8 - x_{15}) \sqrt{4l_2^2 r_2^2 - R_5^2}} \right] \\
 \beta_{2R} &= \tan^{-1} \left[\frac{(x_8 - x_{15}) R_5 + (z_8 - z_{15}) \sqrt{4l_2^2 r_2^2 - R_5^2}}{(z_8 - z_{15}) R_5 - (x_8 - x_{15}) \sqrt{4l_2^2 r_2^2 - R_5^2}} \right] \quad (4.9)
 \end{aligned}$$

ここで、

$$\begin{aligned}
 r_1 &= \sqrt{x_8^2 + z_8^2} \\
 R_1 &= l_2^2 + r_1^2 - l_3^2 \\
 R_2 &= l_3^2 + r_1^2 - l_2^2 \\
 r_2 &= \sqrt{(x_{15} - x_8)^2 + (z_{15} - z_8)^2} \\
 R_4 &= l_3^2 + r_2^2 - l_2^2
 \end{aligned}$$

$$R_5 = l_2^2 + r_2^2 - l_3^2 \quad (4.10)$$

である。また、前頭面内の傾き (α) については、式 (4.9) の計算の前に足首と股部の位置座標から容易に計算できる。なお、上記の計算は始めに α を求め、 α に関する回転変換を施した後に行う。

ZMP軌跡については、ロボットの支持脚の足底接地面によって形成される安定領域内に、いくつかの基本点列を与えこれを適当な方法で補間することによって作成する。補間方法にはスプライン関数、準エルミート補間、線形補間によるもの及びそれらに局部的スムージングを施した場合を設定した。なお、前後の1歩を考慮し連続的な軌跡を作成する。

(c) 補償動作の計算

(b)で作成した歩容とZMP軌跡を入力とし、このZMP軌跡に追従させるための補償動作として、上体の矢状面内の動き ϕ と前頭面内の動き θ を求める。なお、昨年度までに開発した上肢有りモデルでは、両脚の前頭面内の動き α を補償動作の一つとしているが、本年度開発したモデルでは同じ自由度を上体に設けており、この動作 (θ) で補償が可能であるため、ここでは $\alpha = 0$; 一定として解析を行う。このため、ここでのモデルは、足首部のロール軸のアクチュエータを外して考える。但し、股部のアクチュエータについては、今後の方向転換自由度の付加を考え、回転は行わないがそのままとする。

式 (4.3) に関して言えば、未知の運動 $\{\Xi\}_U$ が θ , ϕ であり、既知な駆動力 $\{M_0\}$ が ZMP におけるモーメントの X 成分と Y 成分が 0 であるという拘束条件となる。式 (4.3) は非線形の連立常微分方程式であり、基本歩行パターンの他に θ , ϕ の角度と角速度を初期値として与える必要がある。この入力初期値の推定に関しては、定常歩行を取り上げ歩行動作の周期性と 1 歩ずれた左右対称性から、1 歩の始めと終わりで θ , ϕ の角度と角速度が一致するという境界条件を設定し、これを満足する初期値を求める。正確には回転の方向を考慮して以下の式に従う。

$$\begin{aligned} \theta \Big|_{t=0} &= -\theta \Big|_{t=T/2} \\ \dot{\theta} \Big|_{t=0} &= -\dot{\theta} \Big|_{t=T/2} \\ \phi \Big|_{t=0} &= \phi \Big|_{t=T/2} \\ \dot{\phi} \Big|_{t=0} &= \dot{\phi} \Big|_{t=T/2} \end{aligned} \quad (4.11)$$

T : 1 歩行周期 (2 歩)

すなわち、補償動作に関する周期解を求めることになる。

入力初期値から 1 歩後の最終値の決定には、入力初期値と式 (4.3) より得られる角加速度とを用いて、ルンゲ・クッタ、アダムス・モルトン法を併用して、数値計算により行っている。1 歩分の計算の後境界条件のチェックを行い、境界条件が満足されなければ傾斜法または局所的方法を用いて入力初期値を修正し、境界条件を満足する初期値が見つかるまでこの処理を繰り返す。この数値計算法及び初期値の修正法については文献 2) に詳しい。

4.2.4 補償歩容の計算法（歩行容補償型モデル）

このモデルは、定常歩行における任意の上体動作に対する補償を歩容で行うものである。しかし、足の動作による補償を考えた場合、足の動きは当然のことながら移動の役割を果たす必要があり、要求された歩行速度を保ち簡易な障害物を回避（跨ぎ越え）したりすることが可能でなければならない。しかも、任意に動作させた場合には構造物（床面）やロボットの他の部分との干渉の問題や補償モーメント不足の問題も生じてくる。また、遊脚の接地時刻・位置のみの変更による補償も考えられるが、動作変更にもなう衝撃力の影響やロボットの動作のレスポンスの問題など、直接の補償動作として用いるのは困難である。このような理由から、足は設定した通りに動作すべきであり、足部の動作によって補償する事は望ましくないと考えられる。このため、本研究では足及び上体の動きは設定パターンで動作するとして、この動作に対する補償を歩容、すなわち、腰の位置を制御することで行う。この計算のフローチャートをFig. 4.6に示す。

(a) 支持足首強度計算、足の軌跡、ZMP軌跡及び上体動作の決定

支持足首強度計算によるリンク径の決定は4.2.3節の(a)で示した方法に従う。足の軌跡については支持脚側は常に全点接地であり、遊脚側は歩行周期、歩幅、両脚支持時間及び遊脚の足の通過点から、4.2.3節の(b)と同様にして作成する。ZMP軌跡についても同様である。また、上体動作については腰の動きとの協調やセンシングのための動作などが考えられるが、今回は簡単のため一定値とした。

(b) 腰の軌跡計算

腰の位置の計算は、脚の軌跡の逆運動学計算を含み、腰の位置に対する運動方程式を直接解くことは非常に困難である。そのため、ここでは簡略モデルと設計対象モデルの2つを作成し2段階で安定歩行パターンを生成した。簡略モデルとは、(a)で作成したZMP軌跡に追従する周期的な腰の動作を計算するためのモデルで、Fig. 4.7に示すように、無質量で伸縮自在の下肢を有する。また、腰は一定高であり下肢の動きは遊脚の足のみを考慮して、他は無視した簡易モデルである。なお、腰の周期解の計算には、4.2.3節の上体補償型モデルの補償動作の周期解を求める方法と同じ方法と用いている。但し、境界条件を以下のように変更する。

$$\begin{aligned}
 x_9 \Big|_{t=0} &= x_9 \Big|_{t=T/2} - s1 \\
 \dot{x}_9 \Big|_{t=0} &= \dot{x}_9 \Big|_{t=T/2} \\
 y_9 \Big|_{t=0} &= -y_9 \Big|_{t=T/2} \\
 \dot{y}_9 \Big|_{t=0} &= -\dot{y}_9 \Big|_{t=T/2}
 \end{aligned} \tag{4.12}$$

T : 1歩行周期（2歩）,

$s1$: 歩幅

(c) 歩容の計算

簡略モデルによって得られた腰の位置から、設計対象モデルの下肢の逆運動学計算を4.2.3節と同様に行い、暫定の歩容を作成する。これでロボット全体の動作がとりあえず決定されたことになるが、この歩行パターンは設定されたZMP軌跡に追従するものではない。そこで、設計

対象モデルにこのロボット全体の動作を入力し、設計対象モデルの ZMP 軌跡を求める。この計算は式 (4.1) の ΔF の成分を $(X_{ZMP}, Y_{ZMP}, 0)$ とおき、 X_{ZMP}, Y_{ZMP} について解けば求められる。この計算された ZMP と設定 ZMP との偏差を求め、Fig. 4.6 に示すように ZMP の偏差が許容値に収まるまで反復計算を行う。偏差の収束については、誤差の時間微分値をフィードバックする学習制御の方法などが考えられるが、ここでは早稲田大学で用いている偏差を直接設定 ZMP 軌跡に加えて再計算を行う方法を用いた⁸⁾。計算は以下のように行う。

$$\begin{aligned}
 f(\text{ZMP}_P) &= \text{ZMP}_{C(1)} \\
 \text{ZMP}_{P(1)} &= \text{ZMP}_P + \varepsilon(\text{ZMP}_P - \text{ZMP}_{C(1)}) \\
 f(\text{ZMP}_{P(1)}) &= \text{ZMP}_{C(2)} \\
 \text{ZMP}_{P(2)} &= \text{ZMP}_{P(1)} + \varepsilon(\text{ZMP}_{P(1)} - \text{ZMP}_{C(2)}) \\
 &\vdots \\
 f(\text{ZMP}_{P(n)}) &= \text{ZMP}_{C(n+1)} \approx \text{ZMP}_P
 \end{aligned} \tag{4.13}$$

$f(x)$: 設計対象モデルの ZMP 軌跡 (ZMP_C) を導く,
 ZMP_P : 設定 ZMP,
 ε : 誤差の増幅率 (定数),
 n : 繰り返し回数

4.2.5 低自由度モデルの計算法

4.2.3 節で示した上体補償型モデルと 4.2.4 節で示した歩容補償型モデルとを統合して、低自由度 2 足歩行ロボット・モデルを構築することが可能である。一般に 2 足歩行ロボットの運動は、矢状面内と前頭面内の動きがお互いに与える影響は小さく、これを無視できる。そのため、これらの 2 面について独立に制御を行うことが多い。本研究でもこれまでにシミュレーションからこれらの影響がほとんど無視できることを確認している²⁾。ここでは、矢状面内の補償動作を歩容補償型モデルをもとに腰の X 方向の動作で行い、前頭面内の補償動作を上体補償型モデルの前頭面内の動作 θ で行う。この結果、直進歩行については下肢のロール軸回りの 1 自由度と上体のピッチ軸回りの 1 自由度が不要になり、下肢 4 自由度、腰 1 自由度の低自由度モデルとなる。特に下肢のロール軸の 1 自由度には 4 つのアクチュエータを必要としていたため、これらを外すことによりかなりの軽量化が行える。

計算は以下の手順で行う。

(a) 支持足首強度計算、足の軌跡及び ZMP 軌跡の決定

これまでと同様にして行う。

(b) 腰の軌跡及び上体動作の計算

この計算は歩容の解析の場合と同様に、簡略モデルと設計対象モデルの 2 つによる 2 段階で行う。なお、ここでの簡略モデルは Fig. 4.7 において、 α と ϕ を一定値 ($\alpha = 0$) としたものである。このモデルにより、腰の X 方向の位置及び上体の Y 方向の位置についての周期解を求める。

周期解の解法はこれまでと同様である。但し、境界条件を以下のように変更する。

$$\begin{aligned}
 x_g \Big|_{t=0} &= x_g \Big|_{t=T/2 - s1} \\
 \dot{x}_g \Big|_{t=0} &= \dot{x}_g \Big|_{t=T/2} \\
 y_{19} \Big|_{t=0} &= -y_{19} \Big|_{t=T/2} \\
 \dot{y}_{19} \Big|_{t=0} &= -\dot{y}_{19} \Big|_{t=T/2}
 \end{aligned} \tag{4.14}$$

T : 1 歩行周期 (2 歩),
 $s1$: 歩幅

(c) 補償動作の計算

この計算は4.2.4節の(c)と同様にして行う。簡略モデルによって得られた腰の位置から、設計対象モデルの下肢の逆運動学計算を行い暫定の歩容を作成する。次に、設計対象モデルにこのロボット全体の動作を入力し、設計対象モデルのZMP軌跡を求める。この計算されたZMPと設定ZMPとの偏差を求め、ZMPの偏差が許容値に収まるまで反復計算を行う。偏差の収束計算法についても4.2.4節と同様にして行う。

4.3 シミュレーション結果

4.3.1 基準データの決定

安定歩行パターンのシミュレーションの前に、同一条件のもとで比較するため必要となるいくつかの基準データを決定する。

(a) リンク径

Fig. 4.1 に示した基準データにおける足首部アクチュエータ上部のリンク径を決定する。リンク直径を30~50[mm]の範囲で変化させ安全率の計算を行った結果をTable 4.2 に示す。なお、足首部の可動角はロール軸まわり-50~50[deg]、ピッチ軸まわり-10~10[deg]として計算を行った。但し、上体補償型モデル及び低自由度モデルでは、可動角がピッチ軸まわりのみでアクチュエータの数が減少するため、Table 4.2 の値よりも若干大きくなる。この結果から、基準データにおけるリンク径を40[mm]とした。

(b) 足の軌跡

足の軌跡作成時におけるスプライン関数の次数は任意に設定可能であるが、一般に同じサンプル点を補間する場合、次数が高くなるにつれ振幅が大きくなり、サンプル点近傍では滑らかな動きになる。ここでロボットの機械モデルについて考えてみると、ロボット機械モデルにはどうしても部材のたわみやモデルの誤差などが生じてしまうために、接地時に衝撃力を伴うことが多い。この点を考慮すれば、この影響により設定した時刻より早く遊脚が接地して衝撃力を生じてしまった場合、接地してしまった時刻における設定上の足の動き、すなわち床面との相対速度・加速度の小さいものであれば衝撃力は小さいと考えられる。Table 4.3 は、スプライン関数の次数を3~7次の範囲で変化させた場合の、遊脚接地時の1ステップ前での足の位置・速度・加速

度について示したものである。Table 4.3から分かるように、速度・加速度を減少させるためには次数を上げればよいが、当然のことながら同時にZ方向の高さも減少する。これは、遊脚接地点がサンプル点の一つになっているためである。このZ方向の高さが低くなると、逆に部材のたわみなどで衝撃力が容易に生じてしてしまうことになる。また、遊脚接地時の接地足の床反力についても、スプライン関数の次数が高い方が僅かであるが小さくなる。これらの結果を考慮して、足の軌跡のスプライン関数による補間の次数は5次とした。

(c) ZMP軌跡

ZMP軌跡の作成には、その軌跡を滑らかなものとするためにいくつかの補間方法を設定した。しかし、スプライン関数や準エルミート補間による場合に、歩行周期によっては安定領域ぎりぎりの軌跡となる場合が生じ、特にスプライン関数での補間の場合には補償動作の角加速度が他の場合に比べて大きなものとなった。また、歩行周期ごとに振幅が異なった軌跡となるために同一条件のもとでの補償動作の比較ができないことになる。このため、基本的には基準データとして、サンプル点の補間は線形補間とし局部的にスムージングを施して滑らかな軌跡としたものを用いた。

4.3.2 上体補償型モデル

上体補償型モデルを解析対象とした場合のシミュレーション結果を示す。補償動作の評価は出来るだけ動きの少なくなる場合を最適な動作とし、振幅(θ_{amp} , ϕ_{amp}), $a = \int |\dot{\theta}| dt$, $b = \int |\dot{\phi}| dt$ を規範とし、トルクの評価に角加速度($|\ddot{\theta}|_{max}$, $|\ddot{\phi}|_{max}$)を用いた。

(a) 歩行速度の変更

歩行周期を変更した場合のシミュレーション結果をFig. 4.8に示す。低速になるにつれ補償動作のトルクは減少するが動作角度が増大し、周期解が得られなくなる。これは、低速になるにつれ必要となる動作トルクは減少するが、慣性力による補償効果が減少するため動作が大きくなるのである。この問題は早稲田大学でも低速歩行時の可動角不足⁹⁾として問題となっている。

(b) 上体のおもりの質量及び高さの変更

低速歩行において周期解が得られない場合や可動角不足となってしまうような場合、補償能力の向上には上体のおもりの質量を増加させるか、高さを高くする事が考えられる。これらについてのパラメータ変更によるシミュレーション結果をTable 4.4に示す。ここではリンク径を一定とし、おもりの質量を20~60[kg]の範囲で、胴の長さを500~900[mm]の範囲で変化させた場合の安全率の変化を示している。おもりの質量を変化させた場合は、上記の(a)で得られなかった歩行周期での周期解はやはり得られていない。また、これ以上の質量を増加させることは安全率の面から望ましくない、胴の長さを変化させた場合は、長さが800[mm]以上(Table 4.4のハッチングの部分)で、上記の(a)では得られなかった歩行周期での周期解が得られている。しかも、質量増加が少ないため安全率の低下も僅かである。この結果から、補償能力の向上には上体のおもりの質量を増加させるより、上体(重心)の高さを高く設計することが効果的であり、かつ強度の面からも有効であると言える。

(c) 歩幅の変更

歩幅を 200~400[mm] の範囲で変化させた場合のシミュレーション結果をFig. 4.9に示す。補償動作は歩幅増大（歩行速度高速化）につれ、角度・加速度ともに増大する。ここで、同じ歩行速度（[m/s]）について比較すると、例えば、歩幅200[mm] で1.0[s/step] の場合と歩幅300[mm] で1.5[s/step] の場合など、同じ速度であれば歩幅の小さい方が補償動作が小さくなり（角度・角加速度の両方とも）、ロボット全体の動作角度を小さくできる。これは、狭い場所の通過などの際に有効になると思われる。

(d) 腰のX方向初期位置(X0)の変更

定常歩行において腰の高さ・速度は一定としたが、初期位置については任意である。ここではこの位置の違いによる補償動作への影響について検討する。このシミュレーション結果の一例をFig. 4.10に示す。図から分かるように補償動作評価パラメータが極小となるX0が存在することが分かる。すなわち、最適な腰の初期位置が存在すると言える。また、この位置は歩行周期には関係がなく、歩幅に依存した値であることもシミュレーションにより明らかとなった。

(e) 両脚支持時間 (Td) の変更

歩行の1歩は一方の足のみが接地している単脚支持期と、両足が接地し支持足を切り替える相となる両脚支持期とに分割される。歩容にはこれらの割合は重要なファクターである。Fig. 4.11に両脚支持時間の変更による補償動作のシミュレーション結果の一例を示す。 $|\ddot{\theta}|_{max}$, $|\dot{\phi}|_{max}$, b について極小となるTdが存在することが分かる。すなわち、最適な両脚支持時間が存在すると言える。また、この時間は歩幅には関係なく、歩行周期に依存した値であることもシミュレーションにより明らかとなった。

(d), (e)の結果とも、前頭面内の補償動作 θ への影響は ϕ に比べて僅かであった。

(d), (e)の結果を統合し、上体補償型モデルの安定歩行パターンを作成した一例をFig. 4.12に示す。

(f) アクチュエータの質量と最大トルクとの関係

上記のシミュレーションでは、アクチュエータの質量をとりあえず1個5[kg]としているが、この質量がロボット全体の約50[%]を占めていることから、これの減少による軽量化へのポテンシャルは高い。ここでは、アクチュエータの質量とその時に必要となる最大トルクとの関係をシミュレーションによって検討する。

支持足首におけるリンクの安全率を上体補償型モデルの基準データと同じく5.1程度とし、上体おもりの質量をこの基準データにおけるロボット全体に占める割合と同じ29[%]程度とした。なお、その他の質量及びロボット全体の大きさは変化させない。これらの比較用のロボット・データをTable 4.5に、シミュレーション結果をFig. 4.13, Fig. 4.14に示す。図は、単脚支持期における最大トルク（膝部：膝部が最も大きなトルクを必要とする）及び最大パワーの比較である。この結果は歩幅300[mm]で1.0[s/step]の場合のものであり、実際に適用するには安全を見込んでこれの3~5倍程度、さらに高速歩行させるに至ってはそれ以上の能力を持つアクチュエータが必要である。しかし、現状では軽量で高トルクのアクチュエータは開発レベルではあって

も、製品レベルのものはまだ無く、高トルクを出しやすい油圧アクチュエータや減速機と電動モータとの組み合わせのものが多く、どちらも比較的過重量となる。

4.3.3 歩容補償型モデル

歩容補償型モデルを解析対象とした場合のシミュレーション結果を示す。ここで問題となるのはZMP軌跡の誤差の収束についてであるが、この誤差の評価関数には以下に示す式を用いた。

$$EP = \frac{2}{T} \int_{t=0}^{t=T/2} |ZMP_P - ZMP_C| dt \quad (4.15)$$

T : 1歩行周期 (2歩),

ZMP_P : 設定ZMP,

ZMP_C : 設計対象モデルのZMP

式(4.15)による評価はX、Y方向それぞれ別々に行った。許容値は安定領域のX方向・Y方向それぞれの長さの1 [%]とした。Table 4.6に誤差の収束値を示す。いずれも1.1[mm]以内に収まっておりほぼ設定ZMPと一致した。これは、安定領域内で0.4[%]以下の誤差である。なお、最大誤差が1.0[s/step]の場合においてX方向が8.0[mm]と若干大きくなっているが、この誤差が生じているのは単脚支持期と両脚支持期の接続点であり、ZMPの変化が他のところに比べて急激であるため、設定ZMP軌跡のスージングを変更することによって小さくできると思われる。また、この程度の誤差は実用上何ら問題はない。このシミュレーション結果の一例をFig. 4.15に示す。

4.3.4 低自由度モデル

低自由度モデルを解析対象とした場合のシミュレーション結果の一例をFig. 4.16に、誤差の収束値をTable 4.7に示す。補償動作は同じ基準データにおいて、ほぼ上体補償型モデルの θ と歩容補償型モデルの腰のX方向の位置とを合わせた形となっている。誤差については、歩容補償型の場合より若干大きくなっているが安定領域内で0.5[%]以下の誤差である。

4.4 おわりに

今年度、新たに3タイプの上肢なし2足歩行ロボットの数値シミュレーション・モデルを開発した。本シミュレーション・モデルは、前年度までの問題点を考慮し、よりハードウェア化を意識したモデルであり、ロボット・ボディ・データの変更が容易で、支持足首強度計算、安定歩行パターン生成が行えるなど、自在の解析が可能である。

任意の歩容に対する上体補償動作の解析モデル、逆に上体動作に対する補償歩容の解析モデル及びそれらを統合した低自由度モデルの開発を行い、ハードウェア化についてのいくつかの検討を行った。HASPでは、巡回点検用2足歩行ロボットのハードウェア製作を検討しており、本システムはこのロボット設計に有用となるソフトウェアを目指している。今年度までに平地における直進歩行に関するシミュレーション・モデルの開発及びシミュレーションについてはほぼ完了した。今

後、方向転換動作や外乱を伴う歩行、不整地における歩行について検討を進めていく。

参 考 文 献

- 1) 例えば、浅井，他：原子力知能化システム技術の研究（人間動作シミュレーションプログラム：HASP）－平成2年度作業報告－，JAERI-M 91-101，1991
- 2) 久米：人間型ロボットの二足歩行シミュレーション，JAERI-M 91-047，1991
- 3) 久米：人間型ロボットの二足歩行シミュレーション－シミュレーションモデルの開発－，第9回日本ロボット学会学術講演会，PP.315-318，1991
- 4) 高西，他：2足歩行ロボットWL-10RDによる動歩行の実現，日本ロボット学会誌3巻4号，PP.67-78，1985
- 5) M. Vukobratovic（加藤，山下訳）：歩行ロボットと人工の足，日刊工業新聞社，1975
- 6) 高西，加藤，久米：二足歩行ロボットWL-12(Waseda Leg-12)の設計に関する調査報告，JAERI-M 91-197，1991
- 7) 川路，他：逆運動学を用いた2足歩行ロボットの目標軌道生成アルゴリズム，第6回日本ロボット学会学術講演会，PP.123-124，1988
- 8) 高西，加藤，久米：二足歩行の運動学方程式化－直進における完全歩行の運動方程式－，JAERI-M 90-093，1990
- 9) 高西，他：上体補償機構を有する2足歩行ロボットWL-12の開発，第5回日本ロボット学会学術講演会，PP.579-582，1987

Table 4.1 Recent studies for biped locomotion at related organization. (1/3)

研究機 口ボット名	特 徴											問 題 点 (今後の開発課題)				
	自由度 DOF	重量 kg	身長 cm	歩幅 cm	速度 s/step	ハ 構造部材	ド 7771-1-1	ウ エ センサ	ア センサ	歩 動	行 完全		形 傾斜	感 階段	制 御方式	等
信州大 (H2)	12 片脚:6	9.1	55.6			7ルニクム	DCモータ, SMA7771-1-1 方向 転換用)	外付, 絞リジョク モータ, 71 ガスケット		○		△ 静歩 行, ±5°		方向転換用SMAを用いた7771-1-1の開発		①実際の方向転換歩行は成功していない
熊本大 (H1) SR-3	6 片脚:3			0	0.39 足踏み		DCモータ	0-71イン コダ		○				左右方向の運動を機械的の固有のリズムを利用		①着地時の衝撃が大きい ②リズムのずれ
芝浦工大 (H1) Assby-17	25 片脚:7	250	250							○				床面吸着歩行 床面より電力供給 上体・上肢有り		①歩行用の特別な床が必要
静岡大 (H1)	6 片脚:3	12	66.5	30 25 25 30	1.5 3.3 2.0 2.8		DCモータ ハモニック 付	加速度 モータ, 0- 71インコ ダ, 71 ガスケット		○		○ 上り ○ 下り 5cm ○ 跨ぎ		歩行を2次元に限定, 左右方向は支持機構有り		①矢状面内の運動に限定, 前頭面内の動きは考慮せず
名古屋大 (S63)	6 片脚:3	21	70		0.5					○				歩行を2次元に限定, 左右方向は支持機構有り 足首トルク制御		①矢状面内の運動に限定, 前頭面内の動きは考慮せず ②トルク不足

* 推定値

Table 4.1 Recent studies for biped locomotion at related organization. (2/3)

研究機関名 ロボット	特 徴										問 題 点 (今後の開発課題)		
	自由度 DOF	重量 kg	身長 cm	歩幅 cm	速度 s/step	ハート構造部材	ドウェアセンサ	歩 動 静	完全 定常	行 完全		形 傾斜 階段	制 御 方 式 等
東京電機大 (H1)	10 片脚:5	11.3	98.2	0 35	8 8	アルミ、 CFRP	DCモータ、 エンコーダ	○ ○				重心移動 コントローラ、自己搭載	①傾斜センサによる姿勢制御
東工大 (S62) YANBO	5	10	60		0.7			○ ○	○	○	○	H/C地形(高さの異なる 水平面)用非人間形構造 静⇄動歩容の遷移可能	
機械技研 (H3)	5	27	37	12			DCモータ	○			○	線形倒立振り子モータ 歩行を2次元に拘束	①脚質量が上体質量に 対して十分に軽くなければ ならない ②実時間制御
電通大/岐阜大 (H2) BLR-G2	8 片脚:4	25	97	35~ 40	1.0~ 1.2		DCモータ +非剛性	○	○			角運動量制御型、トルク 分配	
(H3) BLR-G3	10 片脚:5	28	102				ローカル エンコーダ、 傾斜、 角速度、 超音波、 速度、 加速度、 トルク、 床反力	○			○		

Table 4.1 Recent studies for biped locomotion at related organization. (3/3)

研 究 機 関 名	特 徴											問 題 点 (今後の開発課題)		
	自由度 DOF	重量 kg	身長 cm	歩幅 cm	速 度 s/step	ハ 構造部材	ド 駆動方式	エ センサ	ア 動作	行 完全	形 傾斜		態 外乱	制 御 方 式 等
早大 (S63) WL-12R	8 片脚:3 上体:2	99	160	30	1.7	アルミ、 CFRP	電気油圧 駆動方式	圧力センサ、 位置センサ、 角度センサ、 関節トルクセンサ	歩行	○			ZMP規範 上体補償動作型 力制御	①構造部材の剛性不足 ②過重量 ③低速歩行時の上体補償 動作角度不足 ④接地時の衝撃力
(H1) WL-12RⅢ	8 片脚:3 上体:2	100	170	0	0.64 足踏み 1.5			圧力センサ	歩行	○	○	○	未知の外力補償（進行方向 向後方から腰部に作用）	⑤外力作用位置、方向の 変更 ⑥外力補償完全歩行の実 現
(H3) WL-12RⅣ	10 片脚:4 上体:2	120	182	0	2.6 0.64 足踏み				歩行	○	○	○	未知の外力補償（横方向 から腰部に作用） 衝撃緩衝材を用いた受動 回転（roll軸）足底板	

Table 4.2 Factor of safety.

リンク直径 [mm]	安全率
30	2.20
35	3.45
40	5.07
45	7.10
50	9.57

Table 4.3 Ankle's position, velocity and acceleration.

スプライン 関数の次数	X方向 位置 [mm]	X方向 速度 [mm/s]	X方向 加速度 [mm/s ²]	Z方向 位置 [mm]	Z方向 速度 [mm/s]	Z方向 加速度 [mm/s ²]
	3	299.96	179.69	-5156.23	1.97	-114.58
5	299.95	35.88	-2058.93	0.35	-30.06	1668.60
7	299.94	6.68	-575.50	0.06	-6.67	560.44

接地時の0.033[s]前, 300[mm/step], 1.0[s/step], 両足支持時間1歩の20[%]

Table 4.4 Influence of balancing weight's mass and its height to factor of safety.

変更点	リンク直径 [mm]	おもり高さ [mm]	おもり質量 [kg]	安全率
おもり 質量	40	600	20	6.2
			30	5.1
			40	4.4
			50	3.8
			60	3.4
おもり 高さ	40	500	30	5.3
		600		5.1
		700		4.9
		800		4.7
		900		4.6

Table 4.5 Robot bodies data.

アクチュエータ の質量 [kg]	おもり質量 m_w [kg]	リンク直径 d [mm]	ロボット質量 m_t [kg]	質量比 m_w/m_t [%]	安全率
5	30.0	40.0	102.8	29.2	5.1
4	25.5	38.0	87.4	29.2	5.2
3	21.0	35.5	71.8	29.2	5.1
2	16.5	33.0	56.4	29.3	5.2
1	12.0	30.0	40.8	29.4	5.4

Table 4.6 Errors of ZMP trajectory.

歩行周期 T [s]	X _{ZMP} 最大誤差 [mm]	Y _{ZMP} 最大誤差 [mm]	X _{ZMP} E P [mm]	Y _{ZMP} E P [mm]
2.0	8.0	1.7	1.08	0.36
3.0	3.4	1.4	0.69	0.28
4.0	2.8	1.3	0.50	0.18

300[mm/step], 両脚支持時間 1 歩の 20[%]

Table 4.7 Errors of ZMP trajectory.

歩行周期 T [s]	X _{ZMP} 最大誤差 [mm]	Y _{ZMP} 最大誤差 [mm]	X _{ZMP} E P [mm]	Y _{ZMP} E P [mm]
2.0	12.5	1.7	1.43	0.39
3.0	3.6	1.5	0.84	0.30

300[mm/step], 両脚支持時間 1 歩の 20[%]

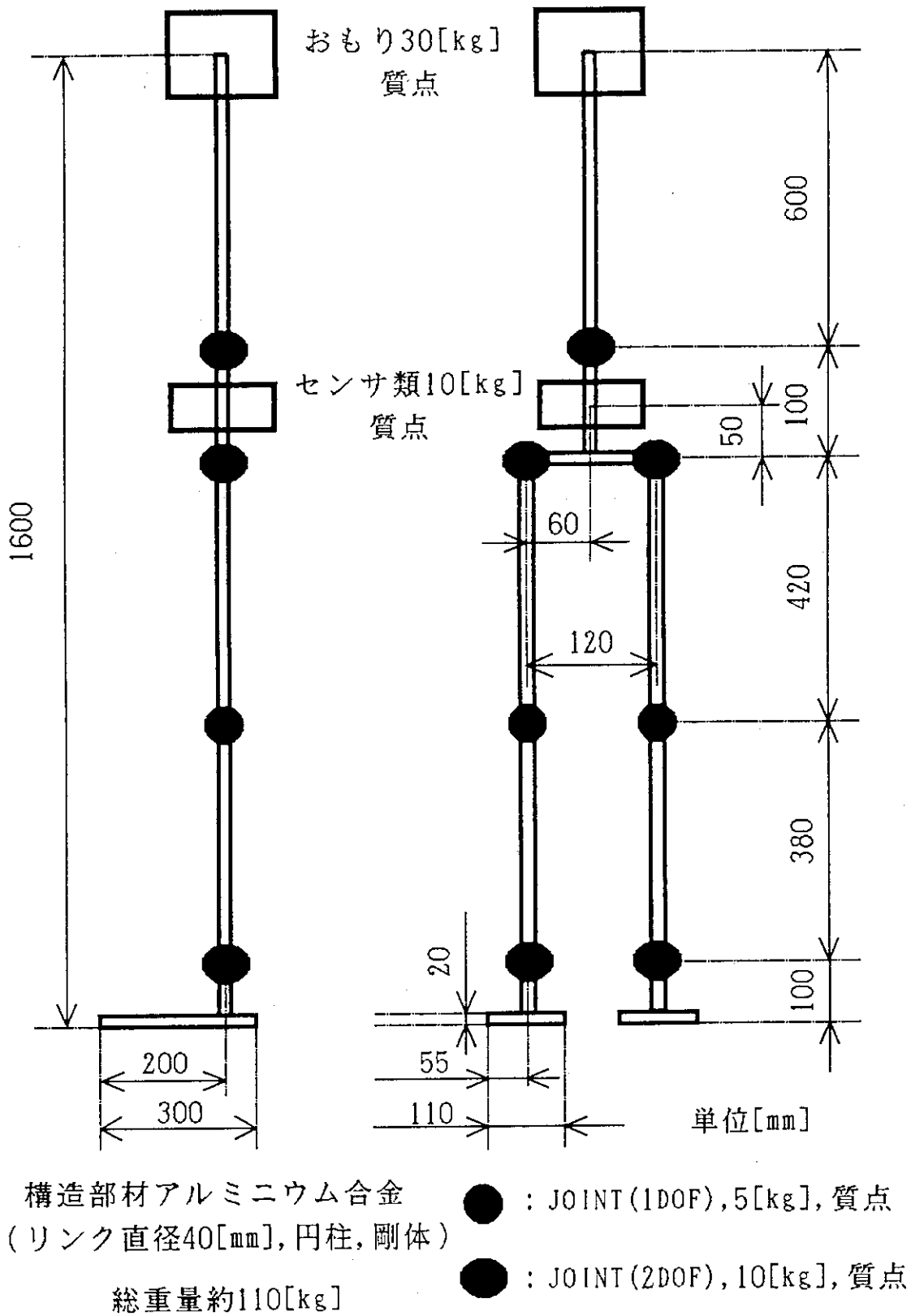


Fig. 4.1 A model of biped locomotion.

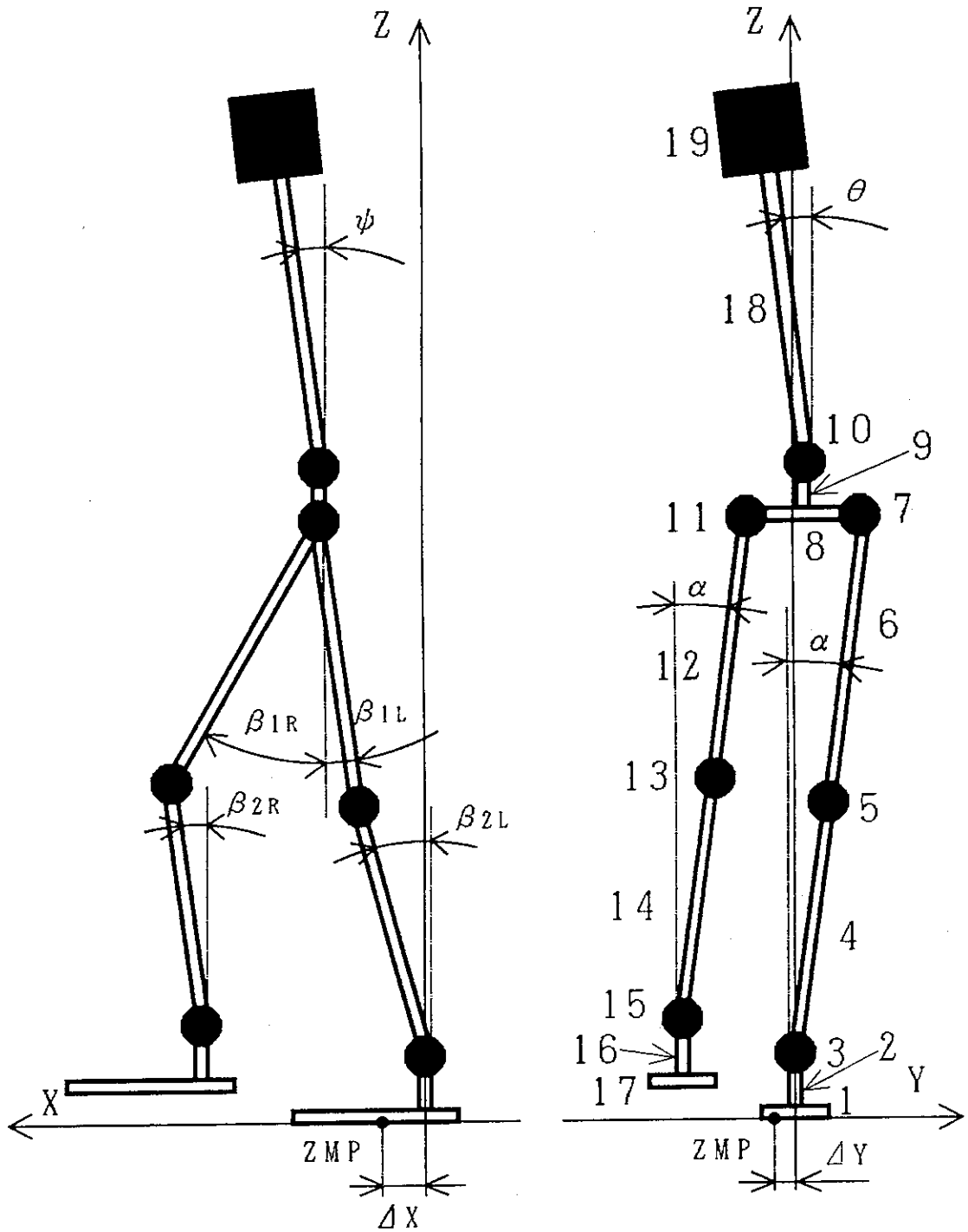


Fig. 4.2 Configuration of robot joints and bodies.

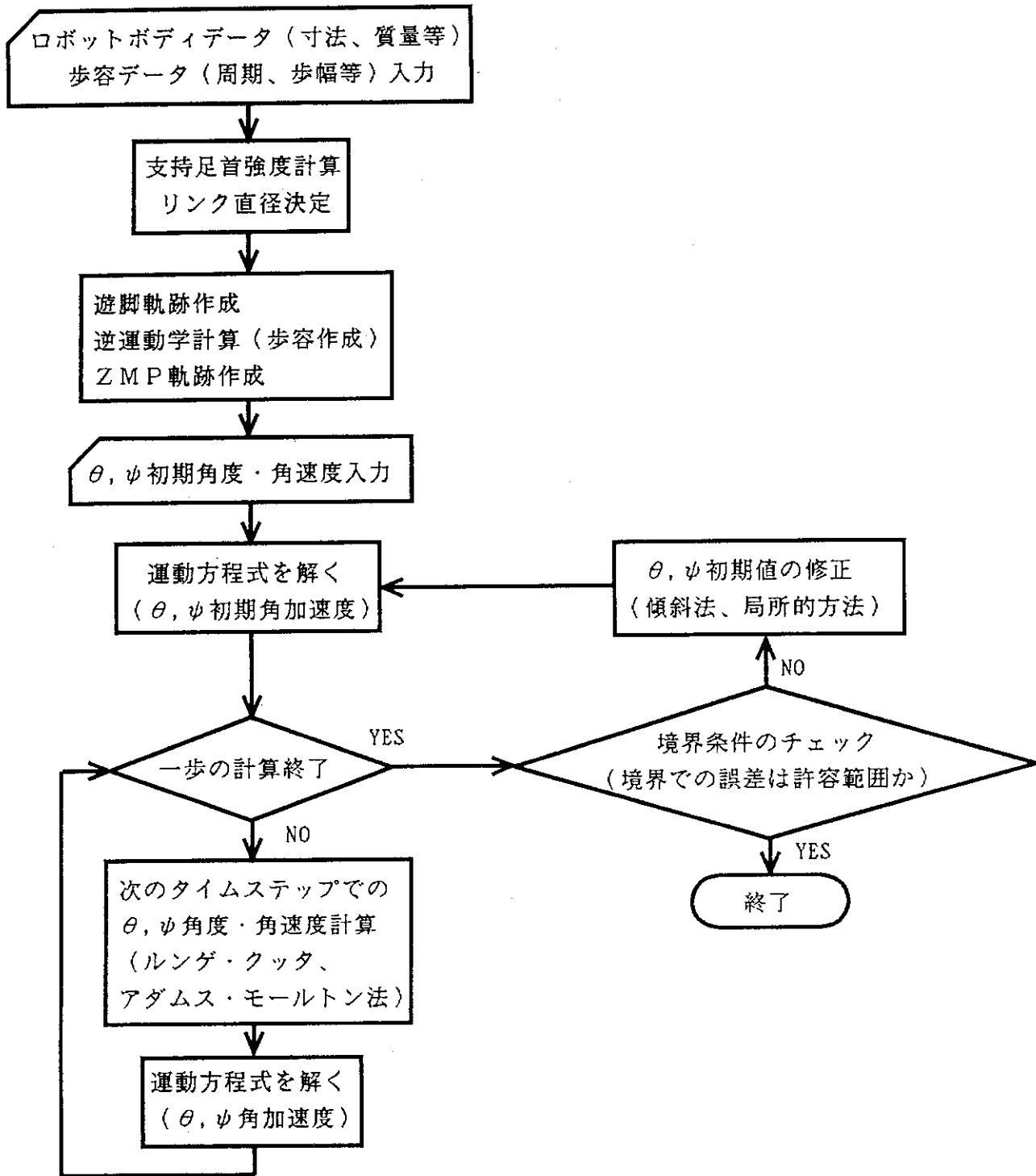


Fig. 4.3 Flow chart to calculate compensative trunk motion.

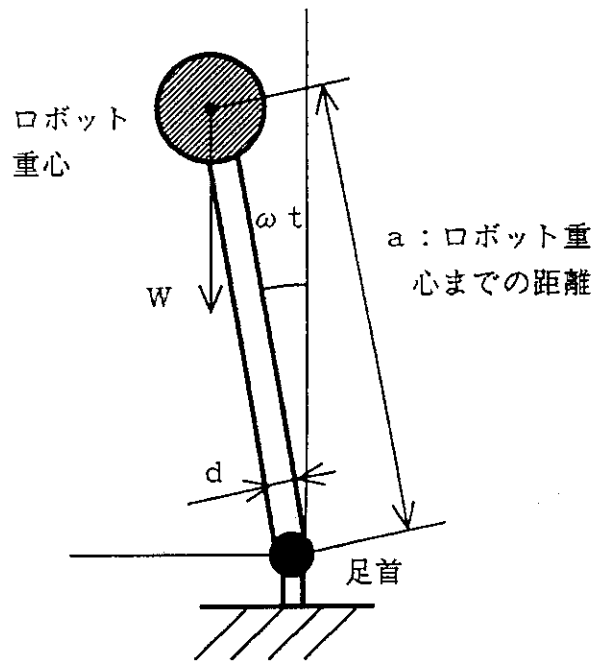


Fig. 4.4 Model for calculation of strength at ankle joint.

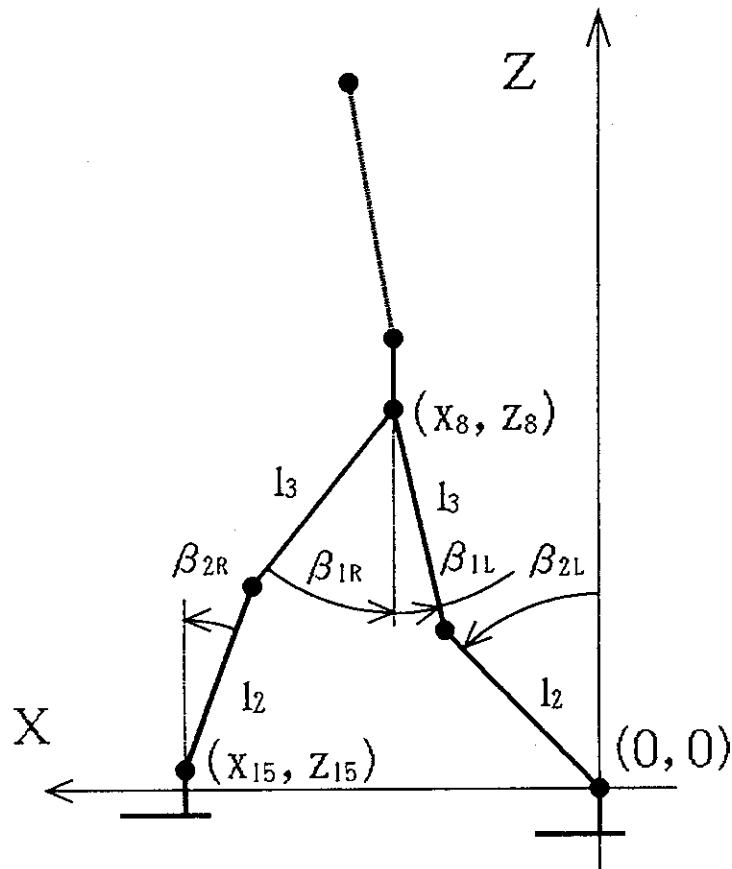


Fig. 4.5 Ankle coordinate system in sagittal plane.

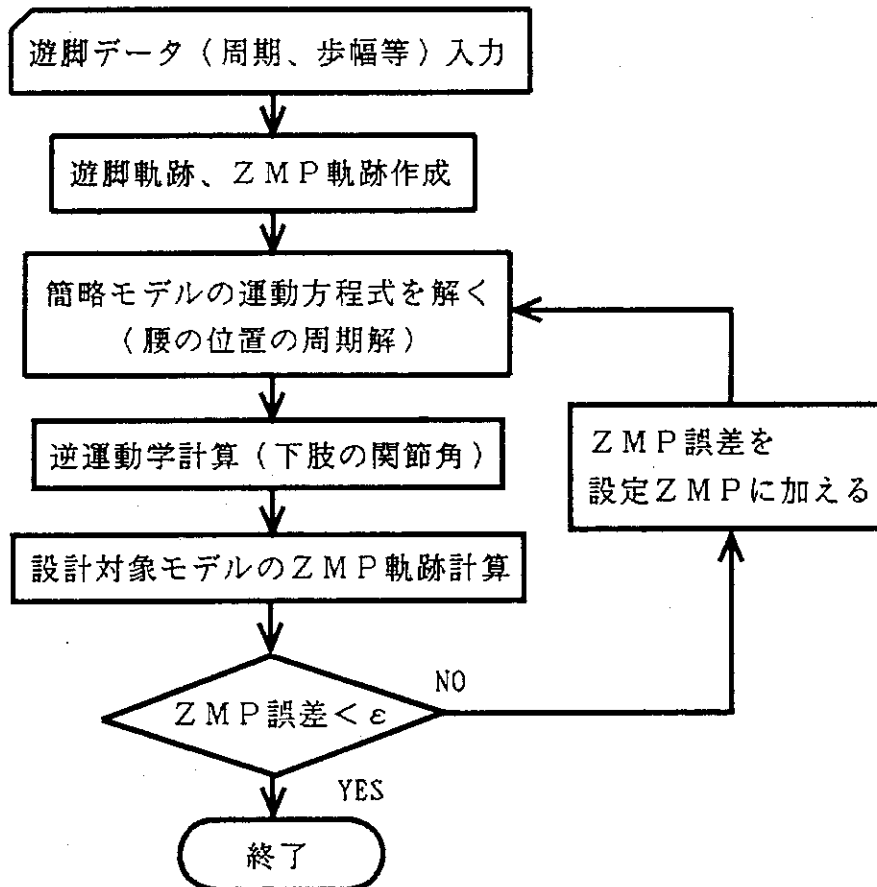


Fig. 4.6 Flow chart to calculate compensative gait.

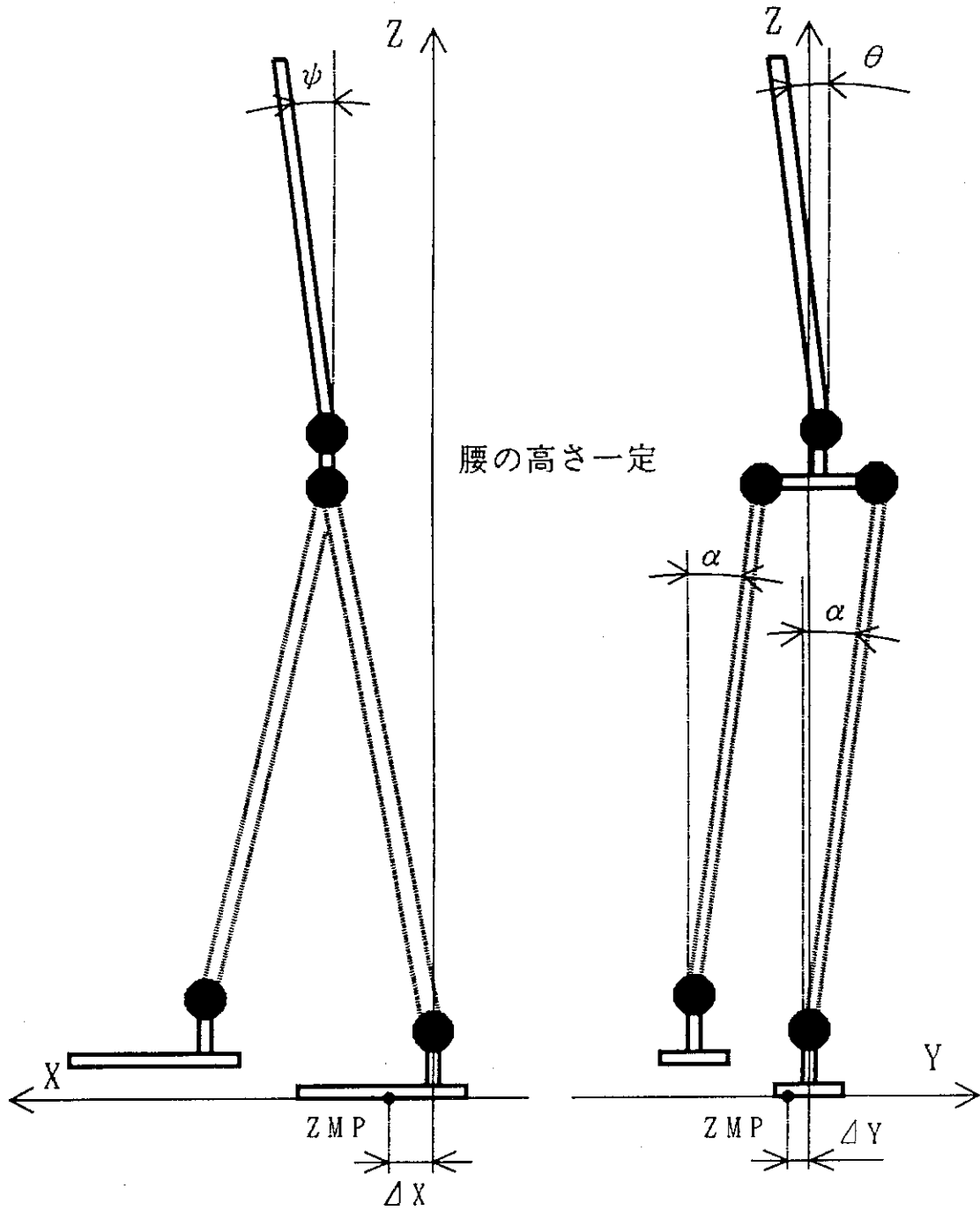


Fig. 4.7 Simplified model of biped locomotion.

300[mm/step], 腰の X 方向の初期位置-100[mm], 両脚支持時間 1 歩の 20[%]

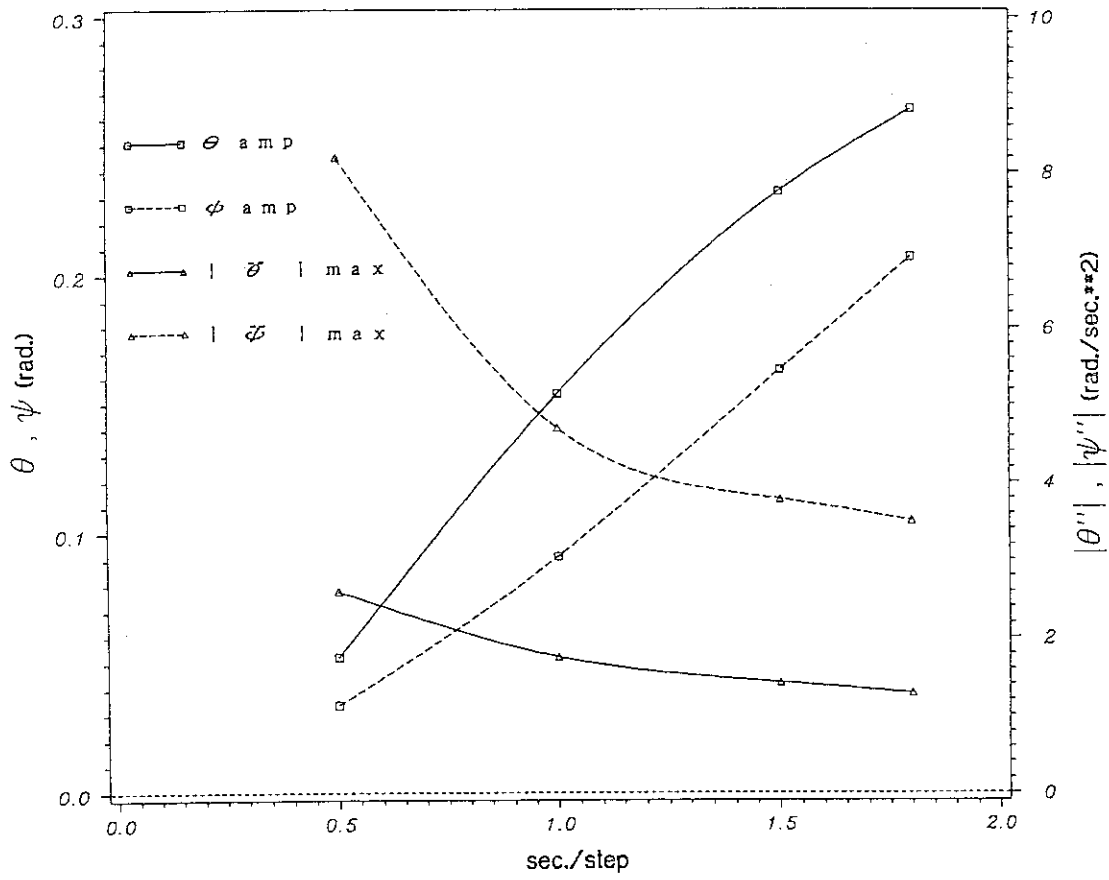


Fig. 4.8 Influence of walking speed to compensative trunk motion.

腰の X 方向の初期位置-100[mm], 両脚支持時間 1 歩の 20[%]

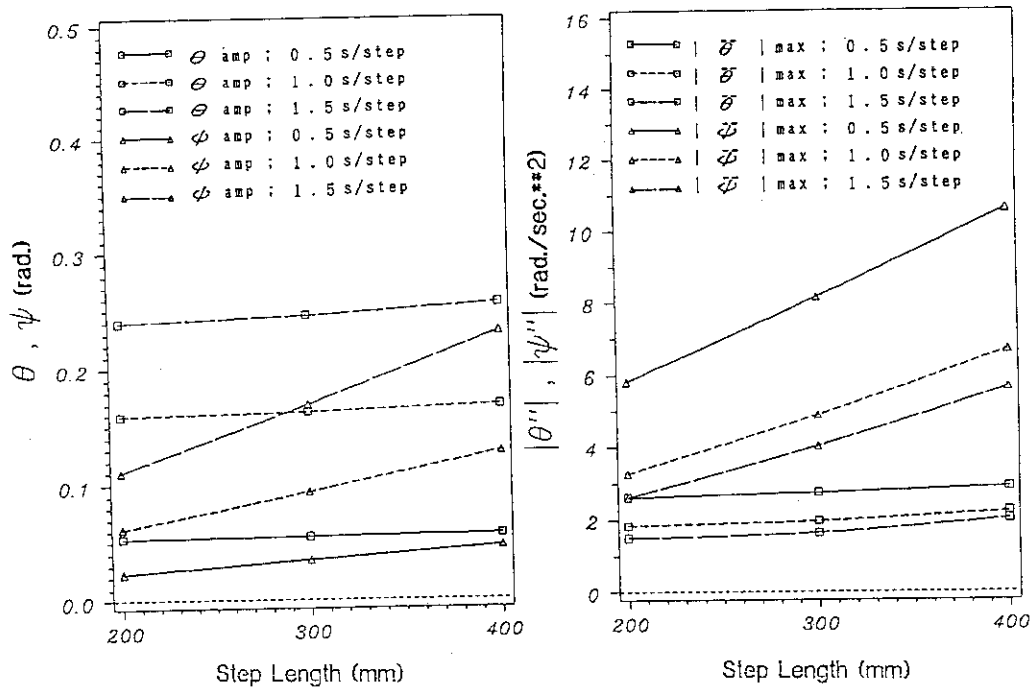


Fig. 4.9 Influence of step length to compensative trunk motion.

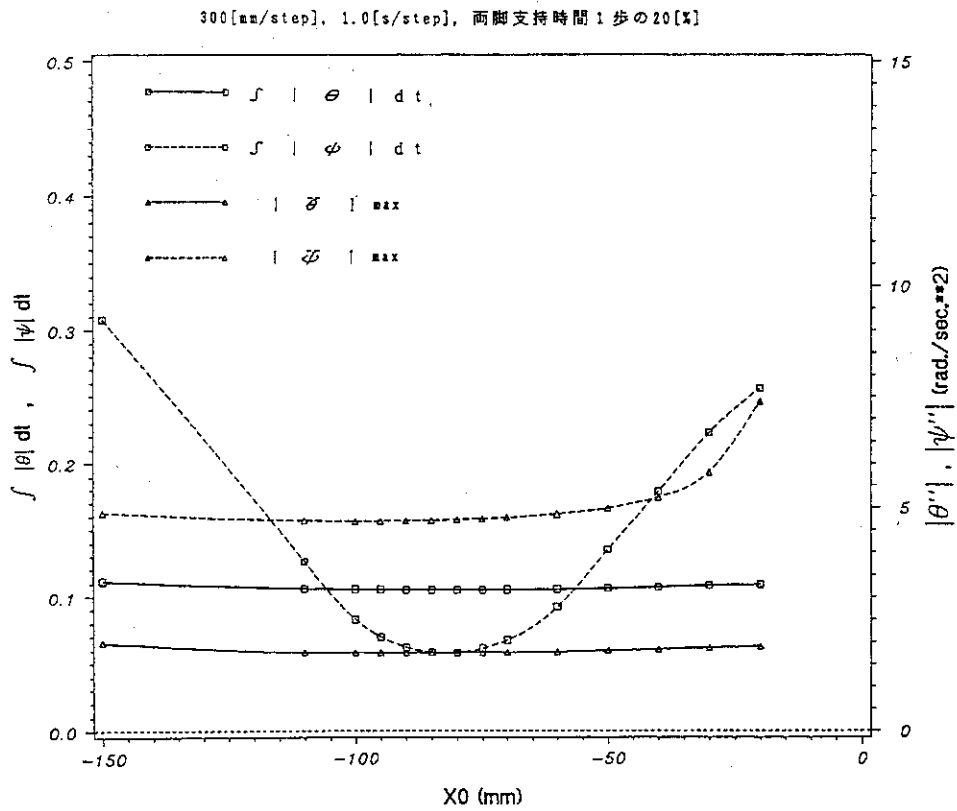


Fig. 4.10 Influence of initial position of waist to compensative trunk motion.

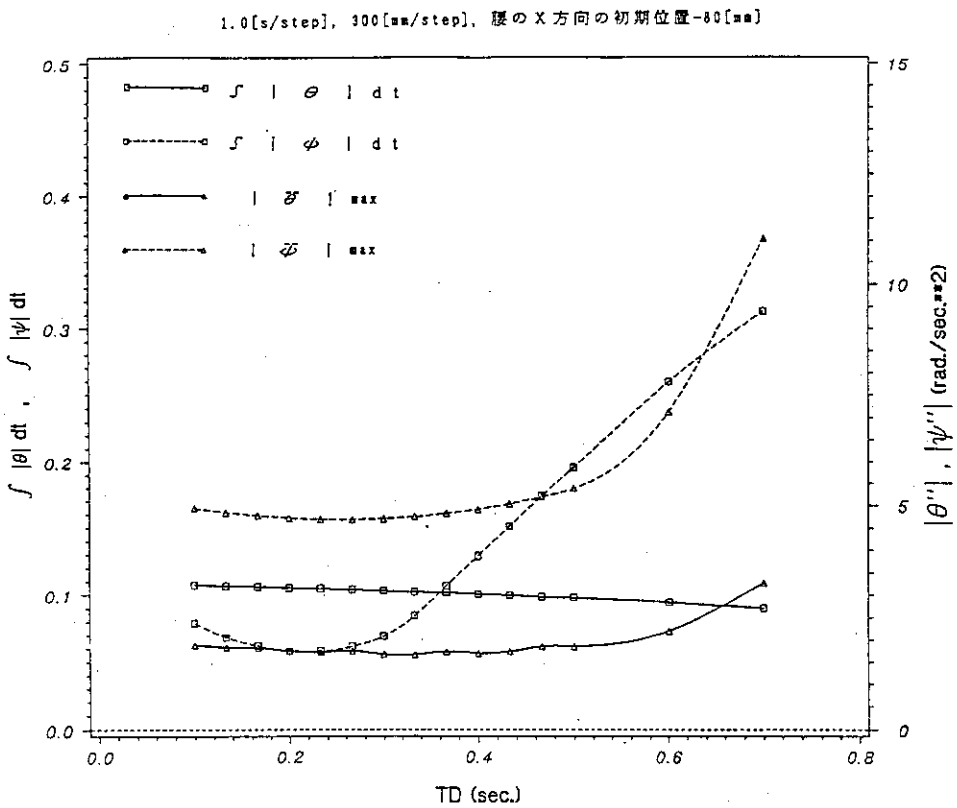


Fig. 4.11 Influence of double supporting time to compensative trunk motion.

300[mm/step], 1.0[s/step], 腰の X 方向の初期位置-80[mm],
両脚支持時間 1 歩の 20[%]

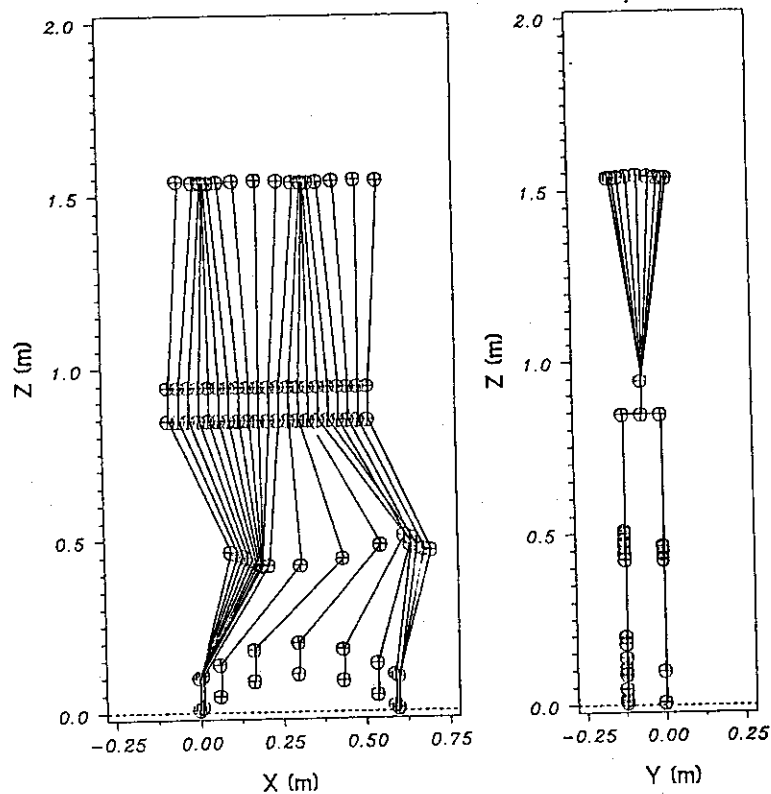


Fig. 4.12 Simulated result of compensative trunk motion.

300[mm/step], 1.0[s/step], 腰の X 方向の初期位置-80[mm], 両脚支持時間 1 歩の 20[%]

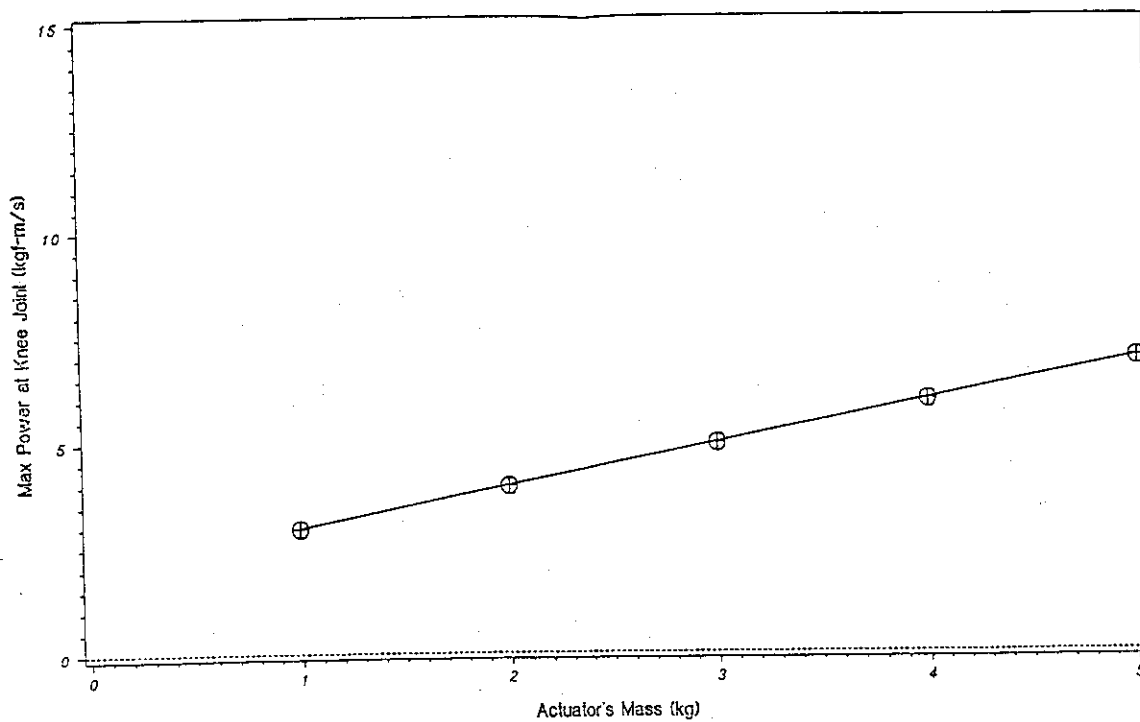


Fig. 4.13 Influence of actuator's mass to maximum torque at knee joint.

300[mm/step], 1.0[s/step], 腰の X 方向の初期位置-80[mm], 両脚支持時間 1 歩の 20[%]

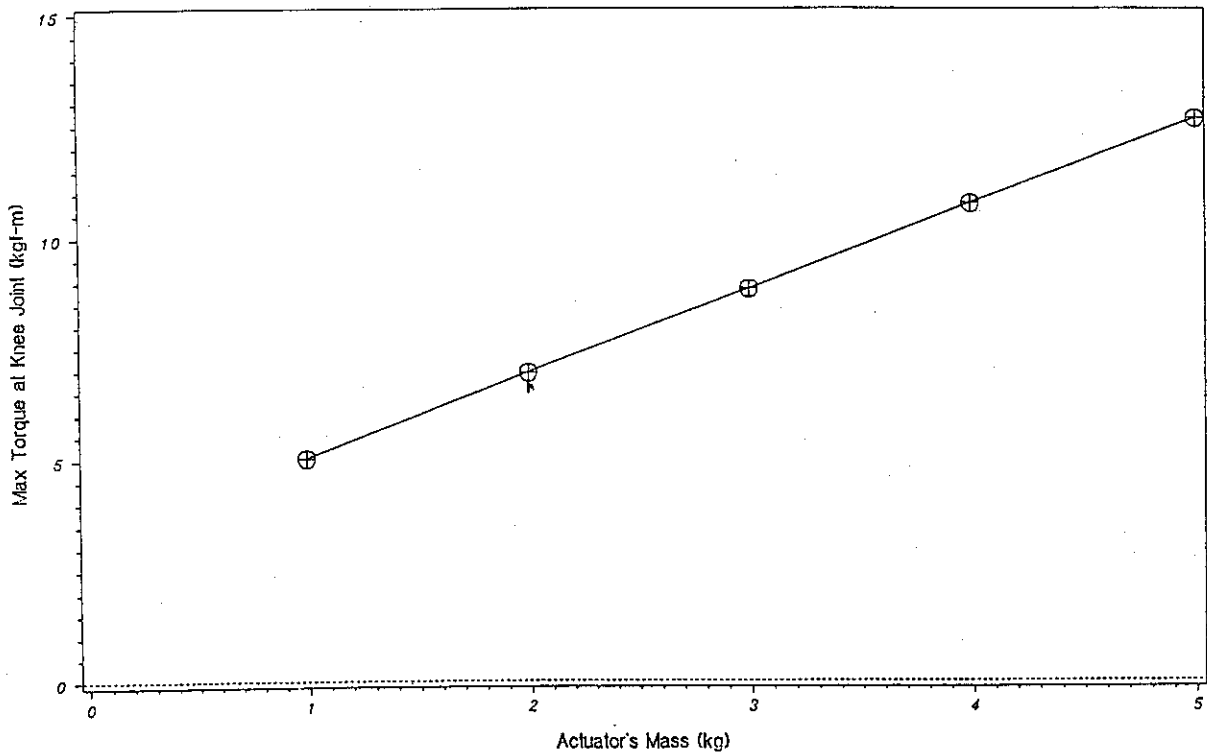


Fig. 4.14 Influence of actuator's mass to maximum power at knee joint.

300[mm/step], 1.0[s/step], 両脚支持時間 1 歩の 20[%]

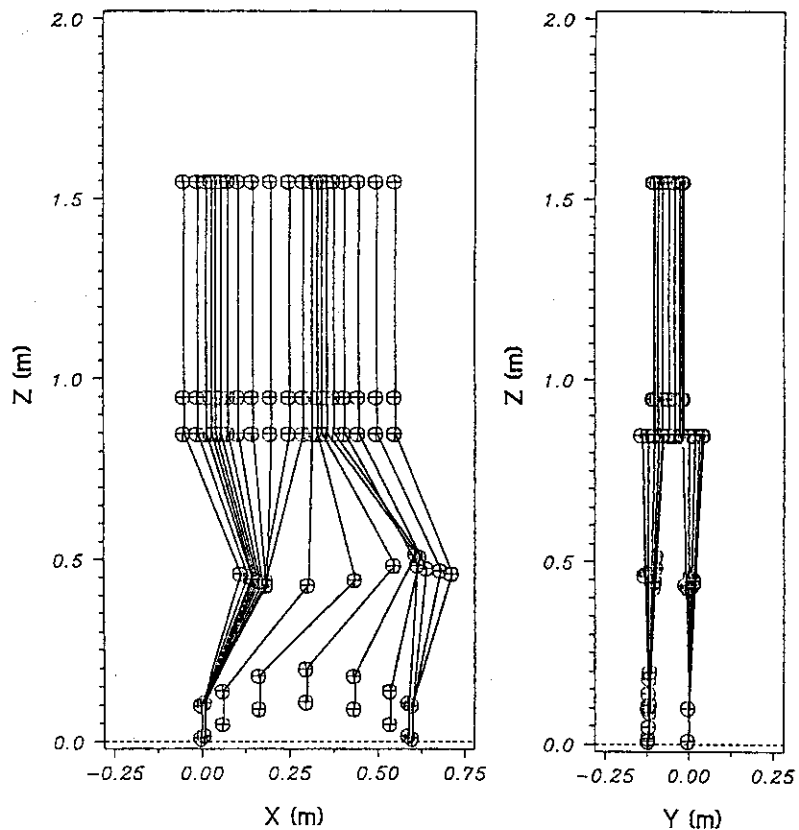


Fig. 4.15 Simulated result of compensative gait.

300[mm/step], 1.0[s/step], 両脚支持時間1歩の20[%]

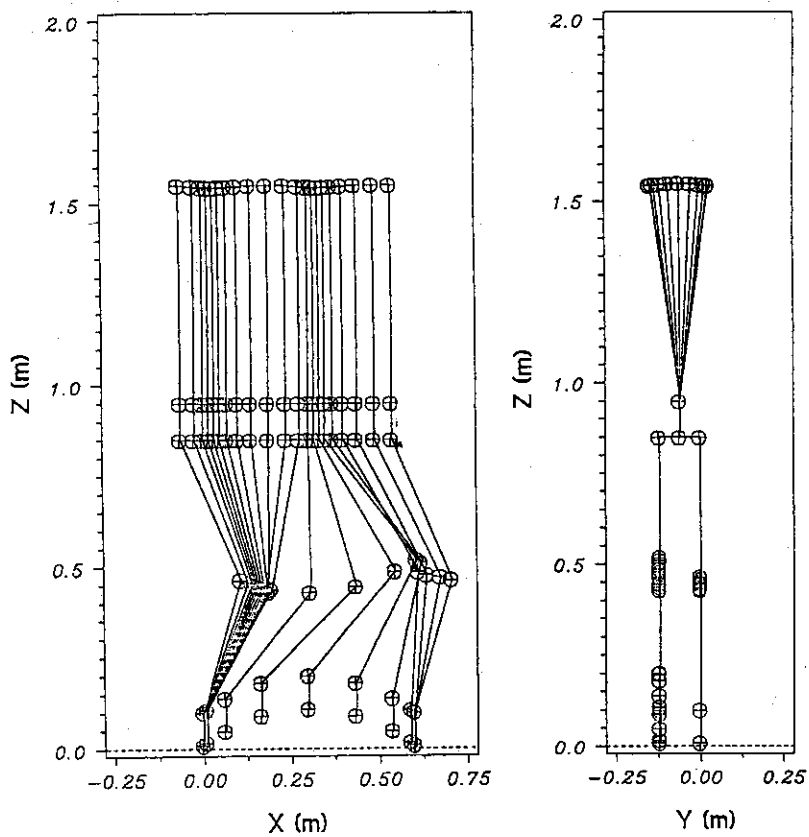


Fig. 4.16 Simulated result of compensative motion. (5-DOF model)

5. 施設形状データベース

5.1 概要

平成2年度までにHASPにおいて構築された施設形状DBは、LISP言語やC言語で記述されており、登録・保存される形状データについてもLISP言語によって参照されていた。LISP言語は、記号処理に適しており、再起的手続きの記述が容易なためプログラムやシステムの開発がしやすい、という長所を持つ反面、FORTRANに比べて数値計算処理が遅い、Garbage Collection (GC) 等実行時の資源消費量が多い、という欠点を持つ。また、視覚認識や被曝線量評価等のHASPにおけるその他のシステムはFORTRAN言語によって記述されており、施設形状DBに保存されるデータはFORTRAN言語による参照に適した簡潔な構造が望ましい。以上の理由から、平成3年度は、施設形状DB上の形状データに関して構造変換を行い、HASPにおけるその他のシステムからの形状データ参照機能を開発・整備した。この結果、新たな施設形状DBがSUN-4上に再構築された。

5.2 旧施設形状DBにおける形状データの構造

平成2年度までの旧施設形状DBは、SUN-3/260Cワークステーション上に構築されていた。世界モデルに属する物体の三次元形状データは、富士通研究所において開発されたモデリング・ソフトFUSIONを使用して登録・保存・参照されていた。三次元物体の表現方法は、Constructive Solid Geometry (CSG) である。CSG手法は、球や直方体等の基本立体(primitive: 以下プリミティブ)の集合演算によって三次元物体を記述する物体表現手法の一つである。CSG手法を採用した理由は、以下のとおりである。HASPにおいては、知能ロボットの動作列を生成する行動計画やロボット躯体に対する放射線損傷度を評価するための被曝線量計算が行われる。この時、三次元物体と直線との交差テストが必要である。つまり、行動計画においては、ロボット躯体と環境中の機器との干渉チェックであり、被曝線量計算においては、モンテカルロ計算における放射線と三次元領域との交差テストである。これらの計算においては、建築物や機器の設計に通常使用されるCADシステム等における境界表現手法による形状データではなく、三次元領域を表現する形状データが適している。

施設形状DBからの形状データの出力例をFig. 5.1は、楕円球や直方体等の9個のプリミティブの集合演算によって表現された三次元物体に関する形状データである。この数値データの説明をFig. 5.2に示す。また、Fig. 5.1に示される数値データに対応するプリミティブの木構造をFig. 5.3に示す。ここで、CSGノードとは、集合演算に使用される演算子である。

5.3 形状データの変換作業

平成3年度の作業において、Fig. 5.1に示される数値データを解釈し、汎用の放射線輸送問題用モンテカルロ・コードMORSEにおける入力データ形式によってプリミティブ（MORSEにおいてはボディ）データ及び論理式を出力するプログラムを作成した。MORSEコードFORMATを用いた理由は、(1)データ構造が簡潔である、(2)旧施設形状DBにおける形状データとの整合性が良い、(3)知能ロボットに対する放射線による損傷度を評価する際のモンテカルロ計算に都合が良い、という点である。変換作業においては、(1)各プリミティブを表現する際、旧施設形状DBにおいて使用された座標系とMORSEコードFORMATにおけるそれとの整合、(2)Fig. 5.1に示される数値データを解釈し、プリミティブの論理式を出力する機能の開発を行った。付録Aに、旧施設形状DB（FUSION系）及び新施設形状DB（MORSE系）における各プリミティブの座標系の違いと変換の内容を示す。また、Fig. 5.4に、新施設形状DBによって出力されたMORSEコードFORMATによる形状データを、Fig. 5.5に、旧施設形状DBによって出力された対応する数値データを示す。これらは、Photo5.1に示されるJRR-3施設内一次区画の一次冷却系主ポンプを表現する形状データである。

5.4 JRR-3 メンテナンス用機器データベースの試作

HASPでは、知能ロボットの動作空間としてJRR-3を想定し、その施設形状データベースをこれまで開発してきた。このJRR-3施設形状データベースは、JRR-3施設の映像化を主目的としてきたため、主要な構造物の3次元形状データを中心に入力されており、各構造物に関するメンテナンス用知識（系統名、点検方法等）や計器等の情報がまだ不十分であった。

ここでは、JRR-3地下1階に設置されている機器のうち、設工認系統図及び機器配置図に記載されている主要機器について、メンテナンスに関連する各種データを収集、整理したので、その概要を記述する。今年度、収集、整理したデータは、地下1階の1次区画、メンテナンスルーム、8区画Iに設置されている主要系統機器である。

データは、機器データと配管データの2つに大きく別れ、機器データには、ポンプ（11）、熱交換機（4）、タンク（3）、盤（6）、ラック（16）、弁（71）、ストレーナ/イオン交換樹脂塔（11）、圧力計（59）、温度計（12）、流量計（26）、液位計（3）、導電率計（6）の合計228件のデータが含まれている。配管データは、1次区画の配管について77名称、434件のデータが含まれている。配管データは、1名称が複数個のデータから構成されている。

機器データベース及び配管データベースについて、それぞれのデータ一覧表（一部）をTable 5.1, 5.2に示す。また、機器データベース及び配管データベースのデータ作成項目をそれぞれTable 5.3, 5.4に示す。各データ作成項目で使用されているコード内容はTable 5.5からTable 5.18に示す。Table 5.1, 5.2の位置座標は、原子炉建屋配置計画図の位置座標によって示されている。即ち、R1-R8が南北の通り芯、RA-REが東西の通り芯で、位置寸法はこれらの通り芯と機器原点までの距離を南方向と西方向をそれぞれ+として示している。また、Table 5.2の配管番号は、配管

呼び径 (mm) , 系統名の略称, 配管番号 (末尾の英字は, 付番である) の順に記載されている。これらのデータは, 計算機で利用可能なように現在データベース化を進めている。

5.5 平成4年度の計画

平成4年度は, ワークステーションによるスタンド・アロン処理を実現するため, ワークステーション上での画像生成機能及び形状データ参照機能を開発する。現在は, SUN-3 (ワークステーション) 上に施設形状DBが構築されており, 画像生成時には, A-50 (ワークステーション), CAPI28 (画像生成用並列計算機) 及び表示用カラーディスプレイを, 動画生成時には, さらにVTR装置等を用いており, 複数の計算機あるいは装置を使用する処理がやや煩雑である。また, 画像生成アルゴリズムとして光線追跡法 (レイ・トレーシング法) を用いているために, 多大の計算時間を要する点も改善する必要がある。平成4年度構築するシステムにおいては, 画像の実時間生成及び動画表示機能を開発するとともに, より使い易いシステムの構築を目指す。このシステムから出力される画像の特徴は, 以下のとおりである。

- ・ 画面解像度: 横512~1024×縦512~1024 (画素)
- ・ RGB解像度: 各8ビット (1670万色)
- ・ 動画表示: 10枚/秒程度
- ・ 形状データ: ポリゴン (三角形)
- ・ 描画アルゴリズム: Zバッファ
- ・ 陰影づけ: Phongのモデルを使用
- ・ アンチエイリアシング機能

画面の解像度の画質に関する量は, 実時間表示機能を重視して決定する。つまり, 画素数 (画面の解像度), 滑らかな曲面を表現するためのポリゴン数, 階段状になってしまった斜め線のぼかしを行うアンチエイリアシング機能等の画質に関する量は, 画像生成時間がそれらに比例して増大するため, 10枚/秒程度の動画表示機能を損なわない範囲において調節する。RGB解像度や描画アルゴリズム及び陰影づけモデルも画像生成時間に関わってくるが, これらの機能や処理速度は, ワークステーションに装備されたハードウェアに依存して決定される。

Table 5.1 A part of JRR-3 facility data table.

No	データ項目	1	2	3	4	5	6	7	8	9	10				11	12			13	14	15	備考	
											設置エリアコード	系統コード	種別コード	機器番号		形状コード	材質コード	点検頻度コード					点検方法コード
平面	断面	形状データの信頼性コード	形状データの信頼性コード	形状データの信頼性コード																			
	機器名称																						
	1次冷却材主ポンプ (No 1)	14	1	1	KP2101	1	1	2	4	1	R3-175/RB-4750	FL-6810	270	0	1	1000	1320	2620	1	3740	1		
	1次冷却材主ポンプ (No 2)	14	1	1	KP2102	1	1	2	4	1	R3-175/RB-2150	FL-6810	270	0	1	1000	1320	2620	1	3740	1		
	1次冷却材補助ポンプ (No 1)	14	1	1	KP2111	1	1	2	4	1	R4-1850/RB-6658	FL-6817.5	180	0	1	650	795	1506	1	683	1		
	1次冷却材補助ポンプ (No 2)	14	1	1	KP2112	1	1	2	4	1	R4-900/RB-6658	FL-6817.5	180	0	1	650	795	1506	1	683	1		
	1次冷却材熱交換器 (No 1)	14	1	2	KX2101	1	1	2	4	1	R4+1200/RB-4550	FL-5252.5	0	0	1	1900	4295	5350	1	36900	1		
	1次冷却材熱交換器 (No 2)	14	1	2	KX2102	1	1	2	4	1	R4+4300/RB-4550	FL-5252.5	0	0	1	1900	4295	5350	1	36900	1		
	原子炉プール溢流ポンプ	14	4	1	KP2501	1	1	2	4	1	R2+1100/RA+6248	FL-7182.5	315	0	1	210	435	595	1	70	1		
	原子炉プール溢流タンク	14	4	3	KT2501	2	1	1	3	1	R2+400/RB-4600	FL-5310	0	+90	1	-	1216	4180	1	1600	1		
	原子炉プール水前置フィルタ (No 1)	14	5	7	KP2601	2	1	2	3	1	R4-2300/RA+2875	FL-6457.5	0	+90	1	-	990	2135	1	360	1	外形寸法は、保護筒	
	原子炉プール水前置フィルタ (No 2)	14	5	7	KP2602	2	1	2	3	1	R4-1100/RB-3133	FL-6457.5	0	+90	1	-	990	2135	1	360	1	外形寸法は、保護筒	
	原子炉プール水浄化ポンプ	14	5	1	KP2601	1	1	2	2	1	R3+1230/RB-6630	FL-7152.5	90	0	1	280	495	690	1	110	1		
	No 1 原子炉プール水イオン交換樹脂塔	14	5	7	KR2601	2	1	2	1	1	R5-2750/RA+1300	FL-5750	0	+90	1	-	1118	3300	1	1400	1		

Table 5.2 A part of JRR-3 pipe data table.

No	データ項目	1	2	3	4		5		6	7	8	9				10	備考
		設置エリアコード	系統コード	配管番号	継手形状コード	継手内容コード	材質コード	点検頻度コード	点検方法コード	位置座標	位置座標	位置座標	位置座標	位置座標	位置座標	位置座標	
	配管名称				始点	終点	始点	終点				始点位置	終点位置	高さ寸法	位置寸法	高さ寸法	位置寸法
	PSS接続配管1	14	1	65A-PCS-32A	2	5	1	KABE	1	2	1	R5+250/RB-5900	FL-7340	FL-7340	R5+0/RB-5900	FL-7340	3
	PSS接続配管1	14	1	65A-PCS-32B	-	3	2	65A-PSS-09	1	2	1	-	-	-	R5+0/RB-5400	FL-7340	3
	PSS接続配管2	14	1	100A-PCS-31A	6	5	1	550A-PCS-04C	1	2	1	R5+250/RB-1300	FL-6750	FL-6750	R5-1500/RB-1300	FL-7340	3
	PSS接続配管2	14	1	100A-PCS-31B	-	5	-	-	1	2	1	-	-	-	R5-0/RB-1300	FL-7340	3
	PSS接続配管2	14	1	100A-PCS-31C	-	3	2	100A-PSS-10	1	2	1	-	-	-	R5-0/RB-3800	FL-7340	3
	減衰タンク出口-PCS主ポンプ入口間配管	14	1	550A-PCS-04A	2	5	1	KABE	1	2	1	R5+250/RB-1300	FL-3901.2	FL-3901.2	R5-600/RB-1300	FL-3901.2	3
	減衰タンク出口-PCS主ポンプ入口間配管	14	1	550A-PCS-04B	-	5	-	-	1	2	1	-	-	-	R5-600/RB-1300	FL-6750	3
	減衰タンク出口-PCS主ポンプ入口間配管	14	1	550A-PCS-04C	-	5	-	-	1	2	1	-	-	-	R3+2080/RB-1300	FL-6750	3
	減衰タンク出口-PCS主ポンプ入口間配管	14	1	550A-PCS-04D	-	7	-	-	1	2	1	-	-	-	R3+2080/RB-6300	FL-6750	3

Table 5.3 Items of JRR-3 facility data.

No.	作成コード項目	備考 (記入内容)
1	機器名称	機器リスト参照
2	配置エリア	コード: 表 5.5 参照
3	系統 (分類)	コード: 表 5.6 参照
4	種別 (設備区分)	コード: 表 5.7 参照
5	物品コード (機器番号)	機器リスト参照
6	形状	コード: 表 5.8 参照
7	材料	コード: 表 5.9 参照
8	点検頻度	コード: 表 5.10 参照
9	点検方法	コード: 表 5.11 参照
10	位置設定コード	コード: 表 5.12 参照
11	位置座標 (平面位置, 断面位置, 設置角度)	
12	形状データの信頼性	コード: 表 5.13 参照
13	形状 (幅, 高さ, 奥行き又は, 外径, 長さ単位; mm)	寸法
14	形状データの信頼性	コード: 表 5.14 参照
15	重量 (単位; kg)	機器重量
16	塗装色	コード: 表 5.15 参照

Table 5.4 Items of JRR-3 pipe data.

No.	作成コード項目	備考 (記入内容)
1	配管名称	機器名称/用途により記載
2	配置エリア	コード: 表 5.5 参照
3	系統 (分類)	コード: 表 5.6 参照
4	物品コード (配管番号)	
5	継手形状	コード: 表 5.16 参照
6	継手内容	コード: 表 5.17, 5.18 参照
7	材質	コード: 表 5.9 参照
8	点検頻度	コード: 表 5.10 参照
9	点検方法	コード: 表 5.11 参照
10	位置座標 (位置寸法, 高さ寸法)	
11	塗装色	コード: 表 6.12, 5.14 参照

Table 5.5 Area codes.

設置エリア	コード	707V ^h M
1 区画	1	FL-7600
¹⁶ N 減衰タンク室	2	FL-7920
照射実験設備室	3	FL-7600
カナル地下室	4	FL-7600
重水区画	5	FL-7860
重水イオン交換塔室	6	FL-7600
計器室 I	7	FL-7600
計器室 II	8	FL-4500
6 区画	9	FL-7600
7 区画	1 0	FL-7600
8 区画 I	1 1	FL-7600
8 区画 II	1 2	FL-4250
メンテナンスルーム	1 3	FL-7600
1 次区画	1 4	FL-7600
炉下室	1 5	FL-7600

Table 5.6 System codes.

系統名称 (系統略称)	コード
1次冷却系 (PCS)	1
2次冷却系 (SCS)	2
重水冷却系 (HWS)	3
原子炉プール溢流系 (POS)	4
原子炉プール水浄化系 (RCS)	5
使用済み燃料プール水浄化系 (SFS)	6
軽水貯留系 (PSS)	7
廃樹脂貯留系 (WRS)	8
冷却系排水系 (WDS)	9
制御棒駆動装置冷却系 (CRS)	10
水平実験孔設備ヘリウムガス充てん系 (HES)	11
計装ラック	12
電気盤	13
その他	14

Table 5.7 Facility codes.

品名	コード
ポンプ	1
熱交換器	2
タンク	3
盤	4
ラック	5
弁	6
ストレーナ/イオン交換樹脂塔	7
圧力計	8
温度計	9
流量計	1 0
液位計	1 1
導電率計	1 2
その他の機器	1 3

Table 5.8 Facility shape codes.

品名	コード	想定する機器
直方体	1	ポンプ、盤類
円柱	2	タンク、熱交換器類
複合物、その他	3	上記で区別できないもの

Table 5.9 Material codes.

材料区分	コード	備考
金属	1	CS/SUS以外で判明したものを付記する。
その他	2	

Table 5.10 Inspection frequency codes.

点検頻度	コード
1回/年 未満	1
1回/年	2
1回/運転サイクル	3
1回/月	4
1回/週	5
1回/日	6

Table 5.11 Inspection method codes.

点検方法		コード
目視		1
分解点検	部分点検	2
	全点検	3
	部分点検 o r 全点検	4

Table 5.12 Location codes.

位置設定	コード	想定する機器
中心位置	1	タンク, ポンプ, 熱交換器
中心及び吐出ノズル位置	2	(今回は該当なし)
前面左端部位置	3	盤, ラック
前面中心位置	4	計器等
通り芯スパン間	5	弁 等
概略の中心位置	6	
概略の前面(左端部)位置	7	

Table 5.13 Reliability codes for facility shape data.

データの内容	コード
図面等により算出（正確）	1
図面等により算出（概略）	2

Table 5.14 Reliability codes for facility weight data.

データの内容	コード
図面等により算出（正確）	1
図面等により算出（概略）	2

Table 5.15 Color codes.

塗装色（支配色）	コード
グリーン (2.5G7/2)	1
ホワイト (N9.5)	2
シルバー	3
その他	4

Table 5.16 Joint shape codes.


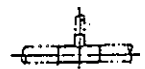

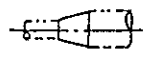
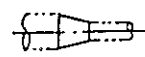
品名	コード	備考
機器ノズル	1	
床、壁	2	
他系統との取合い	3	直線的取合い
配管接続（同一系統内）	4	直線的取合い
エルボ継手	5	曲げ管 
チーズ継手	6	T型継手 
閉塞フランジ	7	止め板 
レジューサ（拡流）	8	相手口径大 
レジューサ（縮流）	9	相手口径小 

Table 5.17 End point codes of joint.

項目	コード	備考
始点	1	継手形状コード1~3,8,9についてのみ、その内容を下表より選択し記載する。
終点	2	

Table 5.18 Description method for joint point.

内容	記載方法
配管接続	配管番号
機器接続	対象機器番号
床貫通	YUKA
天井貫通	TEN
壁貫通	KABE

```

8
9
0 -1 -1
0 1 0
0 2 1
0 2 2
1 2 3
1 2 4
1 2 5
1 1 3
5 2 0
17.5 10.0 10.0 0.0
2 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          0.0          0.0          1.0
8 2 6
10.0 10.0 20.0 40.0
1 0
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          23.9         0.0          1.0
8 1 6
9.9 9.9 20.0 40.04
1 0
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          23.9         0.0          1.0
6 1 5
15.0 15.0 20.4 40.04
1 0
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          34.1         0.0          1.0
0 1 4
18.0 21.0 15.0 40.04
1 0
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          13.4         -7.5         1.0
5 2 7
10.0 10.0 10.0 40.04
1 0
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          23.9         0.0          1.0
6 1 7
10.0 10.0 10.2 40.04
1 0
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          18.8         0.0          1.0

```

Fig. 5.1 Example of output of geometric data from Plant Geometry Database (1/2).

```
5 1 2
8.5 9.9 8.0 40.04
0 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          23.9         0.0          1.0
8 1 1
3.0 3.0 8.0 10.0
2 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
0.0          14.0         0.0          1.0
```

Fig. 5.1 Example of output of geometric data from Plant Geometry Database (2/2).

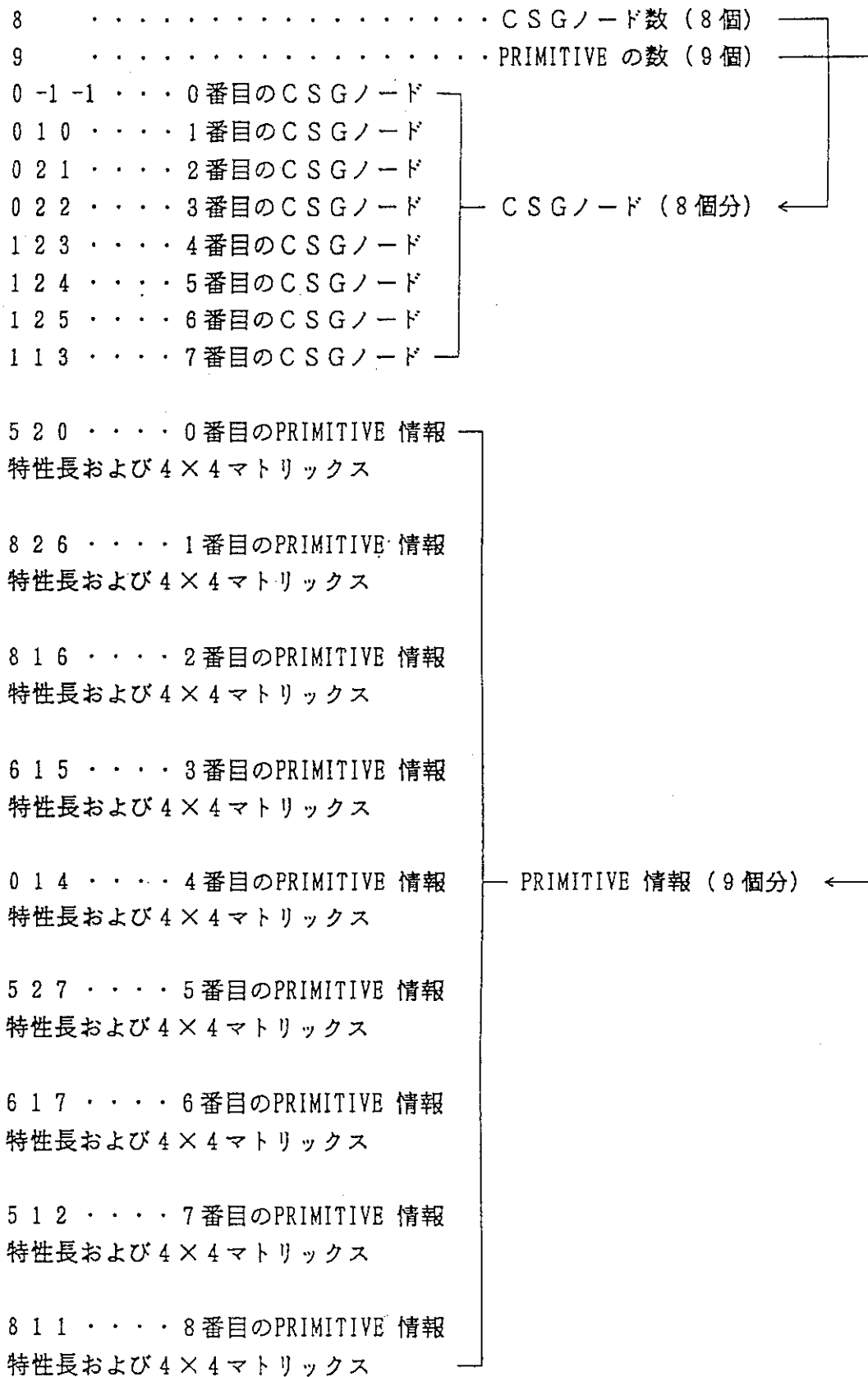
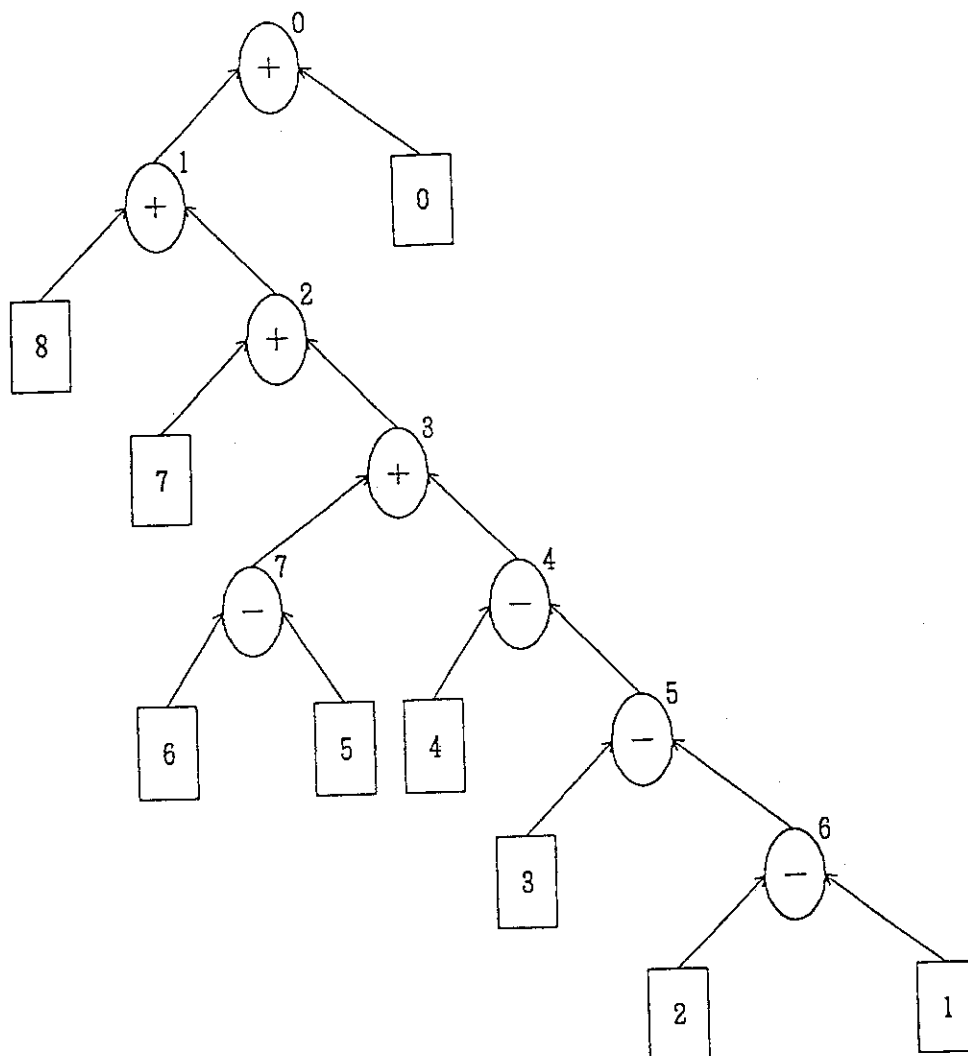


Fig. 5.2 Data format in PG/DB.



※ 単独の数字 (0~7) : CSGノード番号
 四角に囲まれた数字 (0~8) : PRIMITIVE 番号

Fig. 5.3 Structure of primitive tree.


```

0      0      DATA/sectorp/colantmpump1
BOX 1-0.761d+04-0.915d+04 0.000d+00 0.000d+00 0.260d+04 0.000d+00
      -0.128d+04 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.130d+03
BOX 2-0.775d+04-0.905d+04 0.130d+03 0.000d+00 0.240d+04 0.000d+00
      -0.100d+04 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.190d+03
BOX 3-0.775d+04-0.765d+04 0.320d+03 0.000d+00 0.100d+04 0.000d+00
      -0.100d+04 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.145d+03
BOX 4-0.787d+04-0.689d+04 0.465d+03 0.000d+00 0.240d+03 0.000d+00
      -0.153d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.300d+02
BOX 5-0.787d+04-0.765d+04 0.465d+03 0.000d+00 0.240d+03 0.000d+00
      -0.153d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.300d+02
BOX 6-0.789d+04-0.762d+04 0.495d+03 0.000d+00 0.190d+03 0.000d+00
      -0.100d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.355d+03
BOX 7-0.789d+04-0.686d+04 0.495d+03 0.000d+00 0.190d+03 0.000d+00
      -0.100d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.355d+03
BOX 8-0.848d+04-0.689d+04 0.465d+03 0.000d+00 0.240d+03 0.000d+00
      -0.153d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.300d+02
BOX 9-0.848d+04-0.765d+04 0.465d+03 0.000d+00 0.240d+03 0.000d+00
      -0.153d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.300d+02
BOX 10-0.851d+04-0.762d+04 0.495d+03 0.000d+00 0.190d+03 0.000d+00
      -0.100d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.355d+03
BOX 11-0.851d+04-0.686d+04 0.495d+03 0.000d+00 0.190d+03 0.000d+00
      -0.100d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.355d+03
RCC 12-0.825d+04-0.911d+04 0.850d+03 0.000d+00-0.185d+03 0.235d-03
      0.130d+03
RCC 13-0.825d+04-0.905d+04 0.850d+03 0.000d+00-0.600d+02 0.761d-04
      0.250d+03
RCC 14-0.825d+04-0.845d+04 0.850d+03 0.000d+00-0.600d+03 0.761d-03
      0.220d+03
RCC 15-0.825d+04-0.842d+04 0.850d+03 0.000d+00-0.300d+02 0.380d-04
      0.120d+03
RCC 16-0.825d+04-0.781d+04 0.850d+03 0.000d+00-0.605d+03 0.767d-03
      0.900d+02
RCC 17-0.825d+04-0.808d+04 0.850d+03 0.000d+00-0.800d+02 0.101d-03
      0.170d+03
RCC 18-0.825d+04-0.786d+04 0.850d+03 0.000d+00-0.800d+02 0.101d-03
      0.170d+03
RCC 19-0.825d+04-0.661d+04 0.850d+03 0.000d+00-0.120d+04 0.152d-02
      0.350d+03
BOX 20-0.778d+04-0.899d+04 0.320d+03 0.000d+00 0.480d+03 0.000d+00
      -0.940d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.300d+02
BOX 21-0.781d+04-0.897d+04 0.350d+03 0.000d+00 0.430d+03 0.000d+00
      -0.880d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.300d+02
BOX 22-0.786d+04-0.885d+04 0.380d+03 0.000d+00 0.300d+02 0.000d+00
      -0.780d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.480d+03
BOX 23-0.786d+04-0.868d+04 0.380d+03 0.000d+00 0.300d+02 0.000d+00
      -0.780d+03 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.480d+03
TRS 24-0.825d+04-0.875d+04 0.850d+03 0.000d+00 0.190d+03 0.241d-03
      0.200d+03
RCC 25-0.825d+04-0.875d+04 0.850d+03 0.000d+00 0.000d+00 0.570d+03
      0.200d+03
RCC 26-0.825d+04-0.875d+04 0.850d+03-0.570d+03 0.000d+00 0.723d-03
      0.200d+03
RCC 27-0.825d+04-0.875d+04 0.142d+04 0.000d+00 0.000d+00 0.300d+02
      0.250d+03
RCC 28-0.882d+04-0.875d+04 0.850d+03-0.300d+02 0.000d+00 0.380d-04
      0.250d+03
END
      OR    1OR    2OR    3OR    4OR    5OR    6OR    7OR    8OR    9
      OR   10OR   11OR   12OR   13OR   14OR   15OR   16OR   17OR   18
      OR   19OR   20OR   21OR   22OR   23OR   24OR   25OR   26OR   27
      OR    28
END

```

Fig. 5.4 Geometric data in MORSE code format from PG/DB after modification.

```

27
28
0 -1 -1
0 2 0
0 2 1
0 2 2
0 2 3
0 2 4
0 2 5
0 2 6
0 2 7
0 2 8
0 2 9
0 2 10
0 2 11
0 2 12
0 2 13
0 2 14
0 2 15
0 2 16
0 2 17
0 2 18
0 2 19
0 2 20
0 2 21
0 2 22
0 2 23
0 2 24
0 2 25
0 2 26
2600.0 130.0 1280.0 0.0
0 0
1.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 0.0 1.0
-7850.0 65.0 8250.0 1.0
0 1 26
2400.0 190.0 1000.0 0.0
1 1
1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
-7850.0 225.0 8250.0 1.0
0 1 25
1000.0 145.0 1000.0 0.0
1 1
1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
-7150.0 392.5 8250.0 1.0
0 1 24
240.0 30.0 153.0 0.0
1 1
1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
-6770.0 480.0 7944.0 1.0
0 1 23
240.0 30.0 153.0 0.0
1 1
1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
-7530.0 480.0 7944.0 1.0
0 1 22
190.0 355.0 100.0 0.0

```

Fig. 5.5 Geometric data from PG/DB before modification. (1/4)

```

1 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-7530.0      672.5        7944.0        1.0
0 1 21
190.0 355.0 100.0 0.0
1 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-6770.0      672.5        7944.0        1.0
0 1 20
240.0 30.0 153.0 0.0
1 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-6770.0      480.0        8556.0        1.0
0 1 19
240.0 30.0 153.0 0.0
1 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-7530.0      480.0        8556.0        1.0
0 1 18
190.0 355.0 100.0 0.0
1 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-7530.0      672.5        8556.0        1.0
0 1 17
190.0 355.0 100.0 0.0
1 1
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-6770.0      672.5        8556.0        1.0
2 1 16
130.0 185.0 100.0 0.0
2 2
1.2675908E-6  1.0          0.0          0.0
-1.0          1.2675908E-6  0.0          0.0
0.0          0.0          1.0          0.0
-9202.5       850.0001      8250.0        1.0
2 1 15
250.0 60.0 100.0 0.0
2 2
1.2675908E-6  1.0          0.0          0.0
-1.0          1.2675908E-6  0.0          0.0
0.0          0.0          1.0          0.0
-9080.0       849.99994      8250.0        1.0
2 1 14
220.0 600.0 100.0 0.0
2 2
1.2675908E-6  1.0          0.0          0.0
-1.0          1.2675908E-6  0.0          0.0
0.0          0.0          1.0          0.0
-8750.0       849.9995      8250.0        1.0
2 1 13
120.0 30.0 100.0 0.0
1 1
1.2675908E-6  1.0          0.0          0.0
-1.0          1.2675908E-6  0.0          0.0

```

Fig. 5.5 Geometric data from PG/DB before modification. (2/4)

0.0	0.0	1.0	0.0
-8435.0	849.99915	8250.0	1.0
2 1 12			
90.0 605.0 100.0 0.0			
1 1			
1.2675908E-6	1.0	0.0	0.0
-1.0	1.2675908E-6	0.0	0.0
0.0	0.0	1.0	0.0
-8117.5	849.9988	8250.0	1.0
2 1 11			
170.0 80.0 100.0 0.0			
1 1			
1.2675908E-6	1.0	0.0	0.0
-1.0	1.2675908E-6	0.0	0.0
0.0	0.0	1.0	0.0
-8125.0	849.9988	8250.0	1.0
2 1 10			
170.0 80.0 100.0 0.0			
1 1			
1.2675908E-6	1.0	0.0	0.0
-1.0	1.2675908E-6	0.0	0.0
0.0	0.0	1.0	0.0
-7905.0	849.99854	8250.0	1.0
2 1 9			
350.0 1200.0 100.0 0.0			
1 1			
1.2675908E-6	1.0	0.0	0.0
-1.0	1.2675908E-6	0.0	0.0
0.0	0.0	1.0	0.0
-7215.0	849.9977	8250.0	1.0
0 1 8			
480.0 30.0 940.0 0.0			
1 1			
1.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0
-8750.0	335.0	8250.0	1.0
0 1 7			
430.0 30.0 880.0 0.0			
1 1			
1.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0
-8750.0	365.0	8250.0	1.0
0 1 6			
30.0 480.0 780.0 0.0			
2 2			
1.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0
-8835.0	620.0	8250.0	1.0
0 1 5			
30.0 480.0 780.0 0.0			
2 2			
1.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0
-8665.0	620.0	8250.0	1.0
11 1 4			
200.0 190.0 780.0 0.0			
2 2			
1.2675908E-6	-1.0	0.0	0.0
1.0	1.2675908E-6	0.0	0.0
0.0	0.0	1.0	0.0
-8750.0	850.0	8250.0	1.0
2 1 3			

Fig. 5.5 Geometric data from PG/DB before modification. (3/4)

```

200.0 570.0 780.0 0.0
2 2
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-8750.0      1135.0      8250.0      1.0
2 1 2
200.0 570.0 780.0 0.0
2 2
1.0          0.0          0.0          0.0
0.0          1.2675908E-6  1.0          0.0
0.0          -1.0         1.2675908E-6  0.0
-8750.0      850.00037     8535.0      1.0
2 1 1
250.0 30.0 780.0 0.0
2 2
1.0          0.0          0.0          0.0
0.0          1.0          0.0          0.0
0.0          0.0          1.0          0.0
-8750.0      1435.0      8250.0      1.0
2 1 0
250.0 30.0 780.0 0.0
2 2
1.0          0.0          0.0          0.0
0.0          1.2675908E-6  1.0          0.0
0.0          -1.0         1.2675908E-6  0.0
-8750.0      850.00073     8835.0      1.0

```

Fig. 5.5 Geometric data from PG/DB before modification. (4/4)

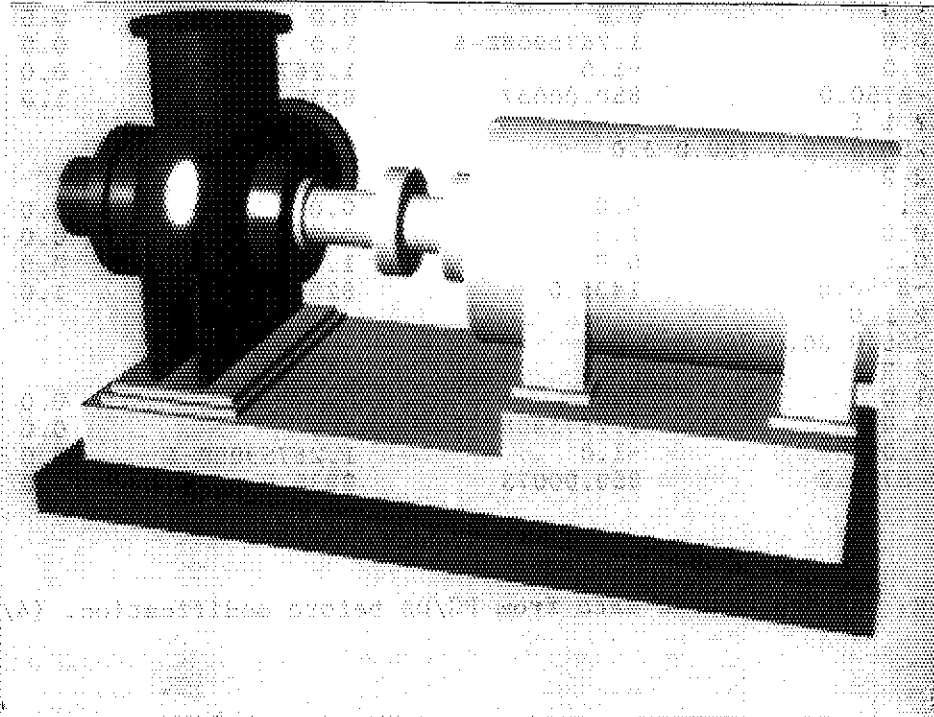


Photo.5.1 3-D image of PCS main pump in PCS area of JRR-3 plant.

6. 被曝線量評価

6.1 はじめに

被曝線量評価システムは、知能ロボットに搭載されたLSI回路等の機器に対する放射線損傷度の評価手法の確立を目的としている。人間型のロボットに対する放射線輸送問題は、極めて複雑な体系に対するモンテカルロ計算が予想されることから、その手法の開発及び妥当性の確認のために、同程度の複雑さを持つ人体に対する外部被曝計算を行った。ここでは、人体模型Cristy Phantom¹⁾及び汎用モンテカルロ・コードMCNP²⁾を用いて、人体に対する実効線量当量の算出を行った。算出値を国際放射線防護委員会(ICRP)の基準データと比較し、手法の妥当性の確認を行った。本章では、線量当量の算出に用いられた計算モデルと計算プログラム、複雑なモンテカルロ計算のために開発された可視化システムの概要、計算結果に対する考察について述べる。なお、より詳しい説明については、文献3)を参照されたい。

6.2 計算プログラム

線量当量の算出には、汎用モンテカルロ・コードMCNPを用いた。HASPは、人体に対する線量当量の算出ではなく、人間型知能ロボットに搭載されたLSI回路等の機器に対する放射線損傷度の評価手法の確立を目的としているため、人体モデルからロボット躯体への形状データの置き換えを考慮して、汎用モンテカルロ・コードを計算に使用した。MCNPコードは、①エネルギーの取り扱いが連続である、②種々の放射線輸送問題を解くための機能が組み込まれている、③種々の断面積データが容易されている等の特徴を持つ。また、米国のロスアラモス国立研究所で開発され、長期にわたって改良を重ねられたプログラムであり、放射線輸送問題に対する完成度が高い。

今回行った人体に対する放射線輸送問題は、計算体系の形状が非常に複雑なために、また被曝線量の算出を効率良く行うために、汎用に開発されたMCNPコードのいくつかの機能を拡張して使用する必要があった。これらのMCNPコードに対する改良は、(1)形状表現用配列の拡張、(2)円環体に対する交差点計算部分の変更、(3)出力プログラムの作成、の3点であるが、その内容については、文献3)において詳細に述べているので、本章では省略する。

6.3 計算モデル

(1) 人体模型Cristy Phantom

人体模型として、米国のオークリッジ国立研究所で開発されたCristy Phantomを用いた。Cristy Phantomは、二次曲面の論理演算で各臓器が表現されている。骨髄は、骨と一様に混じり

合っていると仮定した。骨中の赤色骨髄の重量比をTable 6.1に示す。

二次曲面の論理演算で表現された数学モデルを、MCNPコード用の入力パラメータによって表現するために、変換プログラムを作成した。計算に使用した人体模型Cristy Phantomの各臓器に対する数学モデルとこれに対応するMCNPコード用形状データに関する詳しい説明については文献3)に述べられている。

(2) 実効線量当量を算出する際使用される係数

単一の入射エネルギー E_0 の線源を用いた場合の人体に対する実効線量当量 $DE(E_0)$ は次の式で算出される。

$$DE(E_0) = \sum_i W_i \times \left\{ \sum_j \sum_n K(E_n) \times Q(E_n) \times \Phi_n(E_n) / wt_i \right\} / \Phi$$

ここで、 E_0 は線源のエネルギー、 W_i は*i*番目の臓器または骨に対する荷重係数、 E_n は*i*番目の臓器に対する*j*番目の粒子の*n*回目の衝突時のエネルギー、 $K(E_n)$ はカーマ因子、 $Q(E_n)$ は線質係数、 $\Phi_n(E_n)$ は粒子フルエンス、 wt_i は*i*番目の臓器の重量、 Φ は入射粒子フルエンスである。

・ 荷重係数

荷重係数は、人体全体に対する実効線量当量を定義する際に用いられる係数である。この荷重係数について、ICRP勧告では、「違った臓器に対する放射線被曝による死亡危険性を画一的に評価するための量」と定義している。荷重係数は、全身が均等に放射線被曝した時、その被曝によるある組織の全身に及ぼす統計的危険性の比を表現している。Table 6.2に計算に使用した荷重係数⁴⁾を示す。

・ カーマ因子

カーマ因子は、二次的な荷電粒子の人体に与える影響を考慮するための因子であり、粒子のエネルギーの関数である。人体に対する線量当量は、中性子や γ 線の人体に対する吸収エネルギーではなく、中性子や γ 線と物質(人体)の相互作用によって生成される荷電粒子の初期の運動エネルギーによって線量を決定する。中性子に対するカーマ因子は、人体の軟組織に対して定義されたカーマ因子を文献5)より参照した。使用したカーマ因子をFig. 6.1に示す。 γ 線に対するカーマ値は、カーマ因子等によるカーマ当量への変換を行わずに、単に吸収エネルギーを用いた。

・ 線質係数

線質係数は、放射線の種類によって異なる生物に対する影響を考慮する係数である。荷電粒子が物質(人体)中を通過する際に失う単位距離あたりのエネルギー、つまり線エネルギー付与(LET)によって決まる。ICRPの公刊物によれば、細胞に対する影響の変化及び鼠に対する寿命短縮に関するデータに基づいてこの生物学的効果比は決定されている。カーマ因子と同様、粒子のエネルギーの関数である。しかし、 γ 線の線質係数は、エネルギーの値にかかわらず1とした。中性子に対する線質係数は、カーマ因子と同様、文献5)より参照した。この値は、ICRP Pub. 51⁶⁾において2倍に引き上げられたが、ここでは、それ以前の値を使用している。使用した線質係数をFig. 6.2に示す。

- 放射線場に関する条件

線源は、単一エネルギーの平行ビームである。粒子フルエンスΦで割ることにより、線源の強度に依存しない量、つまり、粒子フルエンスあたりの線量当量が定義される。なお、人体を取り囲む空気の濃度は、標準状態（零度、1気圧）の乾燥空気の 10^{-5} 倍とした。

(3) MCNPコード用入力データ

MCNPコードには、放射線輸送問題において使用されるさまざまなTallyが用意されている。指定された面における中性子を算出するもの、指定された領域における吸収エネルギーを算出するもの等である。今回の計算においては、 γ 線被曝問題に対して、F16 Tallyを中性子線被曝問題に対して、F4 Tally及びF16 Tallyを使用した。F16 Tallyは、指定された領域における単位質量あたりの吸収エネルギーを求めるための γ 線用のTallyであり、F4 Tallyは、指定された領域における平均粒子束を求めるための中性子線用のTallyである。いずれもヒストリーあたり、つまり粒子束あたりの値である。既に述べたように、 γ 線の場合、線質係数は、エネルギーの値にかかわらず1である。このため、MCNPコードによって出力されるrad単位の吸収エネルギーをSV単位の吸収線量に単位変換すれば良い。中性子線による被曝計算を行う場合使用したMCNPコードの機能については、以下のとおりである。

- 二次的な荷電粒子の人体に与える影響を考慮するために、カーマ因子を用いて、中性子束をカーマに変換した。MCNPコードには、粒子束-カーマ間の変換を行う機能が準備されている。この変換機能を使用するために、入力データにおいて、DF_nカード及びDE_nカードを用いた。これらのカードの特徴は、F2やF4カードによって定義された通常のTallyを変更するために、粒子束-カーマ間の修正係数のように点列で表現された関数をエネルギーの関数として入力できる点である。
- 物質との相互作用（核反応）によって二次的に生じた γ 線の人体に与える影響を考慮する必要があるために、中性子線用のTallyだけでなく、プログラムにおいて中性子と共に追跡される γ 線のTallyを用いた。

6.4 モンテカルロ・コード用可視化システムVSMCコード

人体模型を入力するために、可視化システムとして、VSMC (Visualization System for Monte Carlo program) コードを開発した。これは、各臓器の形状が複雑な点を考慮し、作成した形状データの確認を行うため、また、微小な距離によって隔てられた二次曲面間の関係を認識するためである。三次元画像による可視化を行った理由は、複雑な幾何形状を三次元の基本立体または曲面の集合演算によって表現する際、机上の計算で結果を確認することが難しいためである。従来、モンテカルロ・コードの入力データの確認は、そのコードに備えつけのプログラムによって、断面図を線画表示することでなされてきたが、断面の決め方や表示された線の判別が難しく、複雑な三次元幾何形状に対する表示機能が不十分である。今回作成したVSMCコードは、(1)光線追跡法により、立体感のある（つまり色の濃淡の変化がある）三次元画像を生成する、(2)MCNPコードやMORSEコード

に代表される汎用モンテカルロ・コードの入力データをそのまま画像表示用の入力データとすることが可能である、という特徴を持つ。画像生成アルゴリズムや使用方法については、文献7)に詳しい。

6.5 計算結果

使用した計算機資源及び計算体系をTable 6.3に示す。また、Fig. 6.3及びFig. 6.4(a)~(b)に、計算結果とICRPデータとの比較を示す。それぞれ、 γ 線の前方照射、 γ 線の後方照射、中性子線の前方照射及び中性子線の後方照射による外部被曝に対する計算結果の比較である。数値比較によれば、入射エネルギーの高い部分での差異は少なく、低い部分での差異が大きい。入射エネルギーの低い部分（実効線量当量の低い部分）での差異を評価するために、縦軸を対数目盛りにした。 γ 線による外部被曝計算においては、ICRPデータと計算結果が良く一致している。中性子線の場合、低いエネルギーにおいて、計算結果がICRPデータより高めの傾向を示している。

6.6 今後の課題

今回行った人体に対する被曝線量評価手法を、ロボット設計のための要素技術として応用するための今後の課題は以下のとおりである。

(1) ロボット設計への応用可能性

HASPにおいては、知能ロボット設計の際、ロボットに搭載された回路に対する放射線損傷度の評価が必要である。これには、今回行った人体に対する外部被曝計算において確立された複雑形状に対する放射線輸送問題における手法を応用する予定である。この際、人体を構成する組織に対する線量当量評価手法に替わって、ロボットに搭載されたLSI回路に対する放射線損傷度評価手法の開発が必要である。また、ロボット躯体に対する放射線損傷度を評価する際、人体に対する実効線量当量に対応する値を算出する必要がある。人体に対する実効線量当量は、人体を構成する各臓器に対して求めた線量当量と荷重係数を用いて算出したが、ロボットに搭載された回路に対する荷重係数の決定方法も課題である。

(2) 動的な被曝線量評価手法の開発

HASPにおいては、原子力施設内を移動中の知能ロボットに対する放射線損傷度の評価を目標としている。これは、巡回点検等の作業に対するリスクを評価するためである。この手法は、被曝線量評価を行う対象が、ロボットに限らず人体に対しても有用である。

(3) VSMCコード拡張作業

今後のVSMCコード拡張作業において、算出された臓器毎の線量値の程度を色の変化で示す可視化機能を追加する予定である。現在、大型計算機で作成された光線追跡法による三次元画像をワークステーションに転送後、カラーディスプレイに表示しているが、今後の作業によりワーク

ステーション側で画像を生成・表示することを現在検討中である。

参 考 文 献

- 1) Cristy, M. : Specific Absorbed Fractions of Energy at Various Ages from Internal Photon Sources, ORNL-TM-8381, 1987
- 2) Judith, F. : MCNP-A General Monte Carlo Code for Neutron and Photon Transport Version 3A, LA-7396-M, Rev.2 Manual, 1986
- 3) 樋口他, : MCNPコードと人体模型による被曝線量評価システムの開発, JAERI-Mとして投稿予定
- 4) Recommendation of the International Commission on Radiological Protection, ICRP Publication 26, 1977
- 5) Burger, G., et al., Effective Dose Equivalent and it's Relationship to Operational Quantities for Neutrons, European Seminar on Radiation Protection Quantities for External Exposures, PTB Braunschweig, Oct. 13-15, 1980
- 6) Data for Use in Protection Against External Radiation, ICRP Publication 51, 1987
- 7) 樋口他, : モンテカルロ・コード用可視化システムVSMCコード, JAERI-Mとして投稿予定

Table 6.1 Distribution of active marrow in human phantom.

SKELTAL REGION DEFINED IN PHANTOM	DISTRIBUTION OF ACTIVE MARROW IN ADULT PHANTOM
SKULL	8.3 %
SPINE	29.9 %
RIBS	19.2 %
SCAPLAE	2.9 %
ARM BONES (UPPER PORTION)	2.3 %
CLAVICLES	0.8 %
LEG BONES (UPPERPORTION)	3.4 %
PELVIS	33.3 %
TOTAL AMOUNT OF ACTIVE BONE MARROW	1120 g

Table 6.2 The value of weighting factor recommended by the ICRP.

Tissue	weighting factor
Gonads	0.25
Breast	0.15
Red bone marrow	0.12
Lung	0.12
Thyroid	0.03
Bone surfaces	0.03
Remainder	0.30

Table 6.3 Computational resource and calculational parameter.

計算モデル	人体模型：曲面数 200、領域数 110
線源	単一エネルギー：エネルギー範囲10 KeV～ 10 MeV まで、15群（光子被曝） 1 eV～ 10 MeV まで、11群（中性子被曝） 平行ビーム：前方照射及び後方照射の2ケース
サンプリング数	20万ヒストリー
計算時間	・光子被曝計算：5分/群×15群×2ケース＝150分 ・中性子被曝計算：30分/群×11群×2ケース＝660分
使用計算機	FACOM M780/20

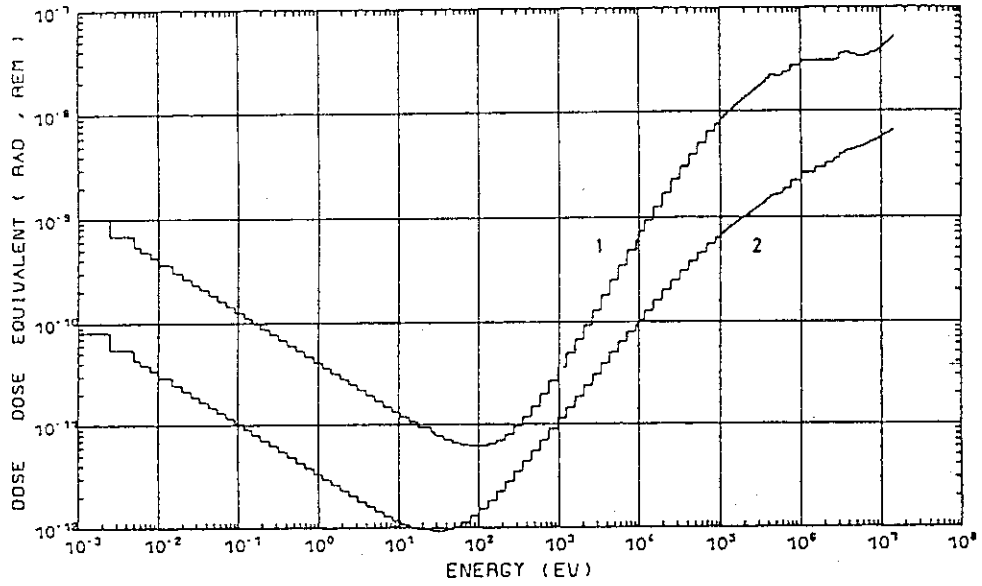


Fig. 6.1 Quality factor-weighted kerma (1) and kerma (2) as functions of neutron energy for soft tissue.

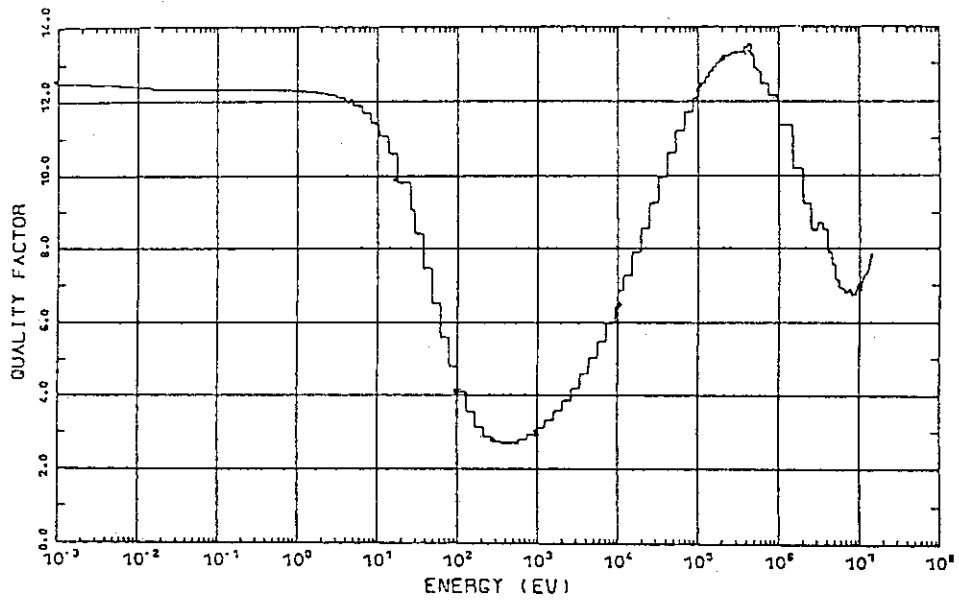


Fig. 6.2 Quality factor for soft tissue as a function of neutron energy.

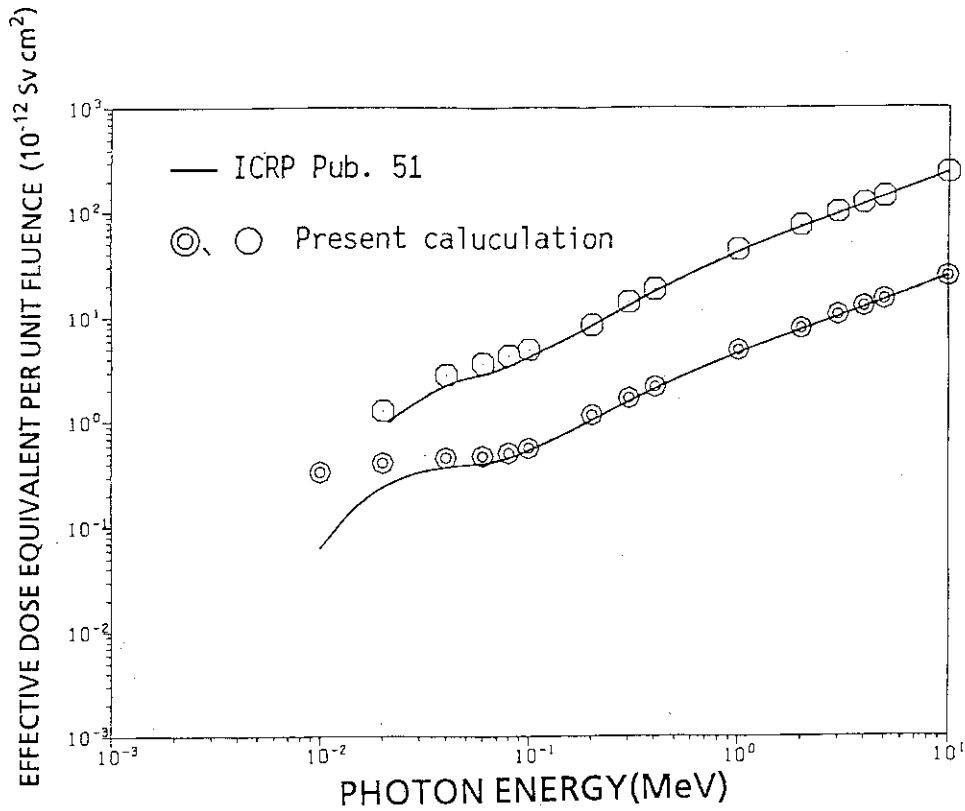


Fig. 6.3 Comparison of effective dose equivalent calculated by MCNP code with the data in the ICRP pub. 51(Photon irradiation).

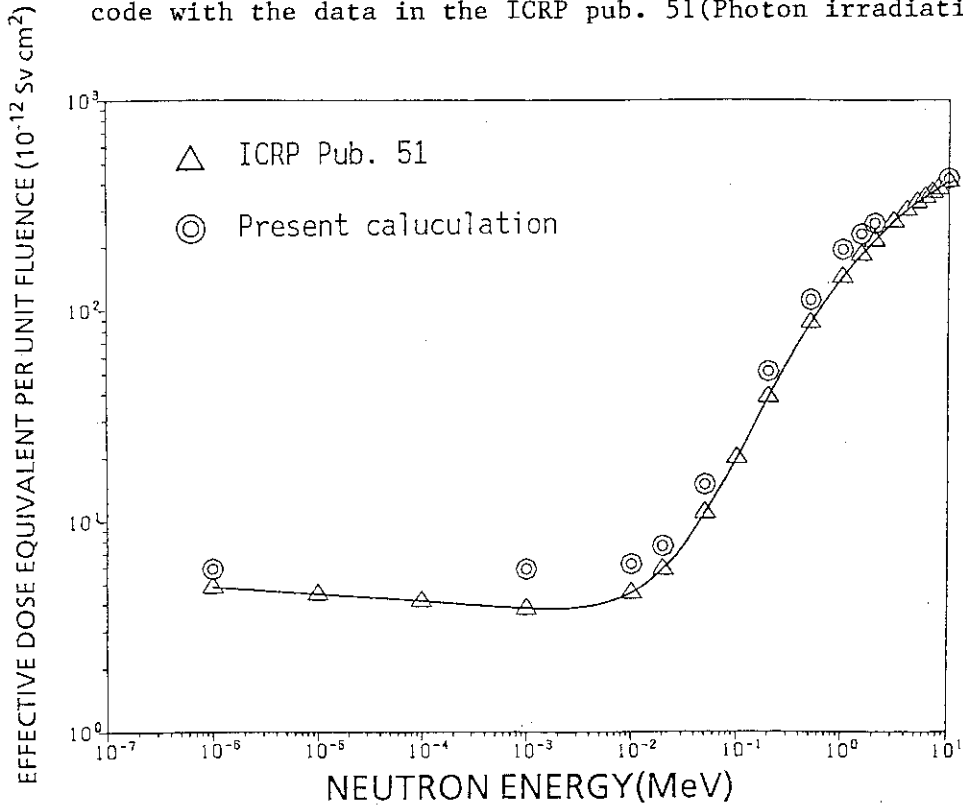


Fig. 6.4(a) Comparison of effective dose equivalent calculated by MCNP code with the data in the ICRP pub. 51(Neutron Irradiation with Anteroposterior geometry)

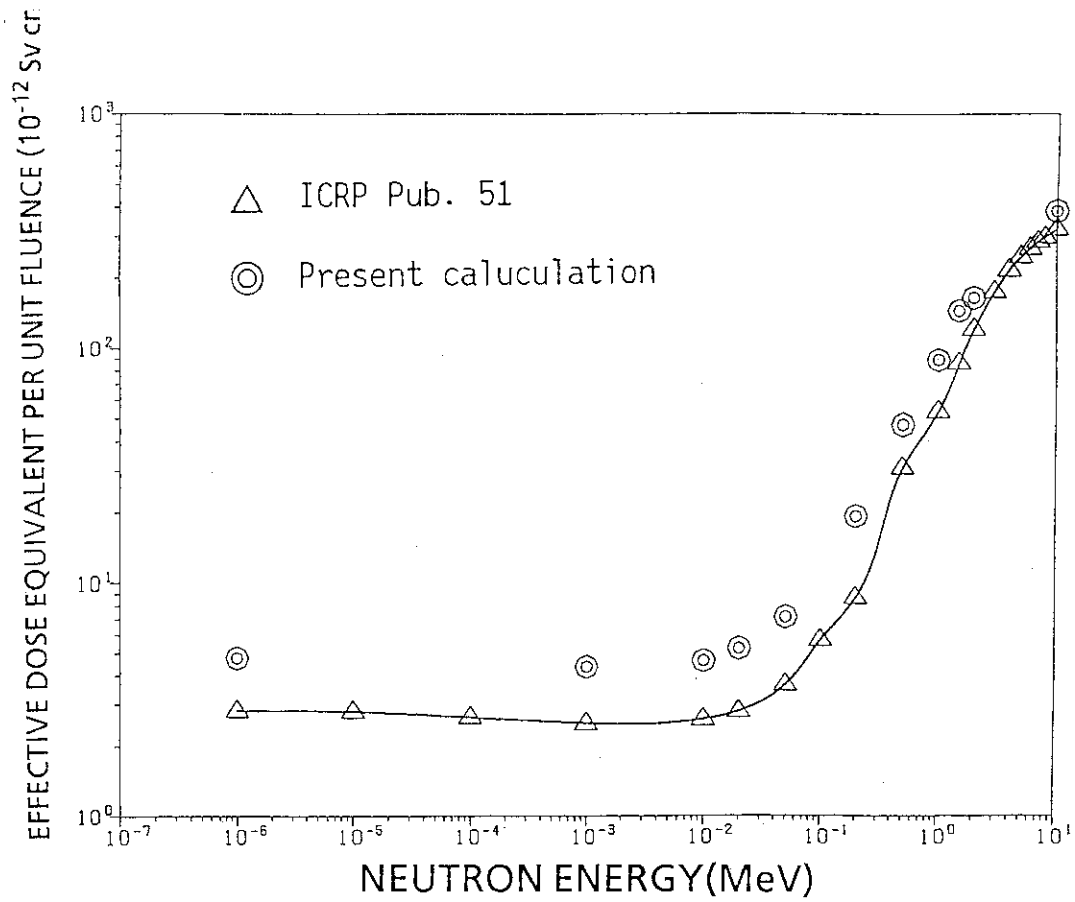


Fig. 6.4(b) Comparison of effective dose equivalent calculated by MCNP code with the data in the ICRP pub. 51 (Neutron Irradiation with Posteroanterior geometry)

7. モンテカルロ計算装置

モンテカルロ計算装置の製作は、昭和63年度の概念設計を経て、平成元年度に詳細設計を終了し、平成2年度より製作段階に入った。平成3年度は、平成4年度末の仮搬入に備えて、MCNPコード他4本のモンテカルロ・コードの整備を行った。これらの整備作業はコードのベクトル化と並列化に分けられるが、ベクトル化作業においては、従来のベクトル計算機による高速化と同様、より長いベクトル長と高いベクトル化率を得るための書き換えを行った他、モンテカルロ計算装置に整備された幾何形状パイプライン、事象分類パイプライン、領域判定パイプライン等のモンテカルロ計算用特殊パイプラインを意識した整備を行った。並列化作業においては、モンテカルロ計算装置において使用可能な並列処理機能を意識して書き換えを行った。モンテカルロ計算装置において使用可能な並列処理機能は、サブルーチン単位に並列処理を行うマクロ・タスク処理とあるサブルーチンにおけるDOループ単位に並列処理を行うマイクロ・タスク処理に分けられ、使用可能なプロセッサ台数は、4台である。

本章では、VP2600を使用したシミュレーションを含む各モンテカルロ・コードの高速化作業について述べる。

7.1 MORSEコードのベクトル/並列化作業

(1) ベクトル化作業結果

① 作業内容

今回、MORSEコード¹⁾のベクトル化作業において行った作業内容は、以下の通りである。

・ コードの解析

VP100用に既にベクトル化済みのコードのVP2600上での振る舞いを調査するため、サブルーチン毎のCPU時間、VU時間、平均ベクトル長、ベクトル化率を計測した。この結果をTable 7.1に示す。この表から、GG、FLUXSTという二つのサブルーチンがベクトル化されておらず、しかも、CPU時間の分布において高い比率を持つ、ということがわかる。

・ 各サブルーチンのベクトル化

今回は、工数の都合上、FLUXSTのみのベクトル化を行った。サブルーチンGGは、サブルーチンG1及びLOOKZから呼び出され、粒子の飛跡と各領域との交差点を計算する。サブルーチンGGをベクトル化するためには、これを呼び出す上位ルーチンがベクトル化されていることが必要である。通常の粒子追跡を行うサブルーチンG1は、既にベクトル化されており、ベクトル版のGGを呼び出すよう書き換え済みである。しかし、ソース粒子の領域番号を決定するために呼び出されるサブルーチンLOOKZは、まだベクトル化されておらず、オリジナル版のGGを呼んでいる。

- ・ 粒子数の増加

より長いベクトル長を得るため、1世代あたりの粒子数を、500粒子から5000粒子と増加した。使用した記憶容量は、それぞれ4.3MBである。この結果、ベクトル長は、増加したが（解析ツールFORTUNEで10倍近いベクトル長を確認）、速度向上率に変化は無かった。この原因は、memory競合と思われる。memory競合とは、プログラムにおいて間接番地により配列を参照する際に発生する負荷である。計算機の主記憶上からベクトル演算を行うベクトル・レジスタヘデータを転送する際、直接番地によって配列を参照する場合は、計算に必要な複数個のデータをほぼ同時に転送できる。一方、間接番地によって配列を参照する場合は、個々のデータを逐次的に転送するために、計算に必要なデータのベクトル・レジスタへの転送終了時間が演算開始時間に間に合わなくなる。つまり、memory競合によって、演算開始時間に待ち時間が生じ、その結果、CPU時間が増大する。このmemory競合は、モンテカルロ・コードのベクトル処理において、常に生じる問題点であり、モンテカルロ計算装置では、この点を改善するために、ロード・パイプラインが強化されている。

- ② 速度向上率

ベクトル処理において得られたオリジナル版に対する速度向上率は、最も大きい値が1.8倍であった。二種類の入力データに対して、1世代あたりの粒子数を変化させて、CPU時間を計測した結果をTable 7.2に示す。

- (2) 並列化作業結果

ある一世代で処理される粒子を分割し、各プロセッサにおいて並列処理を行うため、マクロタスク・レベルの並列化を行った。マクロタスク・レベルの並列処理では、サブルーチン単位で並列処理が行われるために、粒子を発生させてから消滅するまでの処理部分のみをサブルーチン化し、一世代あたり500個の粒子を均等に分割し、各プロセッサにおいて並列処理を行った。書き換えたプログラムを使って、VP2600上でシミュレーションを行い、並列処理による速度向上率を計測した。シミュレーションの結果を、Table 7.3に示す。使用可能なプロセッサ台数を1～4と変化させて、計算時間を評価した。

7.2 VIMコードのベクトル/並列化作業

- (1) ベクトル化作業結果

- ① 作業内容

今回、VIMコード²⁾のベクトル化作業において行った作業内容は、以下の通りである。

- ・ コードの解析

MORSEコードと同様、VP100用に既にベクトル化済みのコードのVP2600上での振る舞いを調査するため、サブルーチン毎のCPU時間、VU時間、平均ベクトル長、ベクトル化率を計測した。結果を、Table 7.4に示す。

- ・ 各サブルーチンのベクトル化

コストが全体の1%以上かつベクトル化率の低いサブルーチンに対するベクトル化を検討した結果、アセンブラ・ルーチンMOREのみのベクトル化を行った。その他のサブルーチンについては、これ以上の効果が見込めないため、実際の書き換えには到らなかった。

アセンブラ・ルーチンMOREは、断面積データを参照するために、二分検索を行う際使用される。FORTRANルーチンに書き換えた後、ベクトル化を行った。この結果、計算時間を一秒（全体の5%）程短縮した。

② 速度向上率

- ・ ベクトル版に関する計算時間短縮：19秒→18秒
- ・ オリジナル版に対する速度向上率：29秒/18秒=1.6倍

(2) 並列化作業結果

MORSEコードと同様、マクロタスク・レベルの並列化作業を行い、VP2600による並列処理のシミュレーションを行い、実効時間を計測した。結果を、Table 7.5に示す。

7.3 MCNPコードのベクトル化作業

MCNPコード³⁾のベクトル化においては、二種類の入力データに対するベクトル化を行った。一つは、中性子に対する遮蔽問題で、円筒状のヘリウム-3 (^3He) 検出器に対するフラックス分布を求めている。もう一方のデータにおいては、 γ 線の外部被曝問題であり、人体に対する線量当量を算出している。以下に、前者の問題に対するベクトル化作業(I)と後者の問題に対するベクトル化作業(II)について述べる。

(1) ベクトル化作業(I)の内容及び結果

① 作業内容

今回、MCNPコードのベクトル化作業(I)において行った作業内容は、以下の通りである。

・ コードの解析

VP100を用いてベクトル化済みのコードのVP2600上での振る舞いを調査するため、サブルーチン毎のCPU時間、VU時間、平均ベクトル長、ベクトル化率を計測した。結果を、Table 7.6に示す。

・ 各サブルーチンのベクトル化

コストが比較的高く、ベクトル化率が低いACETT1, TRNSMI等の6サブルーチンについて、総和計算やサブルーチン・コールを含んでいるためにスカラ処理されていたDOループを分割し、一部をベクトル化した。これらの作業による計算時間の短縮は、すべて0.5秒（全体のCPU時間の4%程度）以下であった。

サブルーチンLGEVALのベクトル化を検討した。LGEVALは、粒子が論理的に正しい領域に存在するか否かの判定を行うサブルーチンである。このサブルーチンは、多数の条件分岐文によって構成されており、しかも、後方向の分岐文が存在するためにベクトル化は、困難であ

る。Fig. 7.1にプログラム・リストを示す。このプログラムに対して、間接番地方式によるベクトル化を行い、「ベクトル長40以上なら、計算時間が短縮可能」という結果を得た。しかし、このサブルーチンは、メイン・サブルーチンからの呼び出しに関してネストの深い位置にあるため、処理粒子数は分散され、小さい（ベクトル長が小さい）。このため、ベクトル化による速度向上は期待できない。但し、このサブルーチンについては、本節(2)で述べるように、モンテカルロ計算装置に装備されたハードウェアによる高速化が可能である。

- ・ 粒子数の増加

モンテカルロ・コードのベクトル処理においては、ある一つの粒子の誕生から死（消滅）までを逐次的に追跡するオリジナル・プログラムのアルゴリズムが変更され、複数の粒子が同時に追跡されるようになる。この処理粒子数を250から500に増加させ、ベクトル長の増大を図った。この作業による計算時間の短縮は、0.4秒程度であった。

② 速度向上率

- ・ ベクトル版に関する計算時間短縮：11.3秒→9.2秒
- ・ オリジナル版に対する速度向上率：13.7秒/9.2秒=1.5倍
- ・ 入力データ：photonのみ。12,000ヒストリー。

(2) ベクトル化作業（II）の内容及び結果

① 作業内容

既に原研情報システムセンターで一部書き換え済みのベクトル版MCNPコードをさらにベクトル化した。ここで使用された入力データは、 γ 線の平行ビームによる人体に対する外部被曝に関する線量当量の計算に用いられるものである。この入力データに対して新たにサブルーチンをベクトル化した。また、この入力データにおいては、1セル（領域）あたりの記述が100 wordsを越えてしまうため、これに対応するMCNPコードの機能拡張を行った。作業内容を以下に示す。

- ・ 形状表現用配列の領域拡張

MCNPコードにおいては、自由二次曲面の論理演算で三次元領域を表現するが、計算モデルを表現する際に使用される三次元領域や二次曲面の数及び一つの三次元領域を表現する際に使用される二次曲面の数に対する制限、つまり、形状表現における制限がある。体系を表現するために使用できる領域（セルと呼ばれる）は、150セル以内であり、また、各セルは100word以内のデータで表現されなければならない。一方、さまざまな臓器や骨によって構成される人体内の形状は複雑であり、今回計算に使用した人体模型を、このMCNPコードにおける制限下で記述することが不可能であった。そこで、形状表現用配列を拡張し、計算に使用した。この拡張作業では、サブルーチンCHKCS及び動的割付による配列格納用コモンEPHCOM及びコモンIMXDCSを書き換えた。その結果、使用可能なセル数及び各セルを記述する際のword数に関する制限は、現在使用しているMCNPコードにおいては無い。

- ・ プログラム動作の変更に伴う新たなサブルーチンのベクトル化

今回、TORUS, COLLIDP, KLEIN, ROTAS, ISOS等のサブルーチンを新たにベクトル化した。

これらのサブルーチンは、これまでの入力データで使用されていなかったもの、使用されていたが、計算コストが低いため、オリジナル版のまま、使用されていたものの2種類がある。いずれも、計算コストの高い順にベクトル化を行い、現在も一部作業中である。

② 速度向上率

現在までに行われた作業による速度向上率を、Table 7.7に示す。処理粒子数は250～2500である。

③ モンテカルロ計算装置による速度向上率の評価

MCNPコードのベクトル処理（処理粒子数；1500）におけるモンテカルロ計算装置による速度向上率について、幾何形状パイプラインと領域検査パイプラインに対する評価を行った。

i 幾何形状パイプライン適用可能部分

- サブルーチンTRACK1(a)：3秒
- サブルーチンCHKCL1(b)：1秒

ii 領域検査パイプライン適用可能部分（サブルーチンLGEVALの計算時間）

- サブルーチンTRACK1(c)：42秒
- サブルーチンCHKCL1(d)：9秒

それぞれのパイプラインによって5倍の高速化が行われたと仮定すると、CPU時間は、

$$A - B + B / 5$$

A；現在のベクトル処理時間：127秒

B；両パイプライン適用可能部分の処理時間： $a + b + c + d = 55$ 秒

となる。つまり、幾何形状パイプライン及び領域検査パイプライン適用時のCPU時間は、

$$127 - 55 + 55 / 5 = 83 \text{ (秒)}$$

が予想される。この時の速度向上率は、MCNPコード・オリジナル版のCPU時間は181秒なので、

$$181 / 83 = 2.2 \text{ 倍}$$

となる。

上記の速度向上率は、プロセッサ1台の場合である。プロセッサを4台使用した並列処理による3倍の速度向上率を想定すれば、モンテカルロ計算装置によるMCNPコードに対する速度向上率は、6.6倍となる。

7.4 GRAPEコードの並列化作業

(1) 並列化作業内容

GRAPEコード⁴⁾については、前年度の作業において、既にベクトル化済みである。この結果を、Table 7.8に示す。今回、さらに高速化を行うために、結晶格子損傷をシミュレーションする際に入射される複数の粒子について並列化を行った。入射粒子が被入射物質を構成するある原子に運動エネルギーを与え、その結果、その原子は格子点から飛び出す。本コードは、その時の、全原子の運動をシミュレーションしている。原子間の力を計算する際、逐次計算においては、前の

時間ステップの原子の位置を保存せず、計算が済み次第、各原子の位置を更新する。並列計算においては、同期をとるために、前の時間ステップの原子の位置のみを力の計算に用いる様プログラムの書き換えが必要である。この並列化においては、D0ループの並列処理を行うマイクロタスク・レベルの並列処理を想定して、書き換えを行った。入力データは、5000evのエネルギーを持つ33,300個の入射粒子を、1/50秒を1タイム・ステップとして1秒間のシミュレーションを行うものである。

(2) 速度向上率

VP2600を使用した並列処理シミュレーションにおいて、ほぼプロセッサ台数に等しい速度向上率を得た。本コードの並列処理における粒度 (granularity) が高い理由は、計算コストが並列化したD0ループに集中しているためである。

参 考 文 献

- 1) M. B. Emmett: The MORSE Monte Carlo Radiation Transport Code System, ORNL 4972, 1975.
- 2) L. J. Milton: VIM User's Guide, Applied Physics Division, ANL, June 1981.
- 3) LASL Group TD-6: MCNP-A General Monte Carlo Code for Neutron and Photon Transport, LA 7396-M, LASL, July, 1978.
- 4) 朝岡, 他: 固体材質の照射損傷計算機シミュレーションの現状, JAERI-memo(所内資料)6076, 日本原子力研究所, 1976年9月

Table 7.1 CPU distribution of MORSE code.

MORSEコードCPU時間分布										ベクトル化版									
オリジナル版					ベクトルモード					ベクトルモード					ベクトル化率				
FORTE 順位	サブ チン名	FORTE コスト(t)	CPU (%)	CPU 時間 (ss)	FORTE コスト(t)	CPU (%)	CPU 時間 (ss)	CPU (%)	CPU 時間 (ss)	VU時間 (ss)	倍率	平均	最長	最長	FORTE	実測			
1	GC	25.0	13.6	376	30.1	28.70	739	17.2	235	71	3.14	92	42	170	84.15	77.80			
2	PIRETA	11.5	8.6	237	2.4	1.94	50	0.6	8	3	6.25	40	10	80	96.81	90.00			
3	RELCOI	11.2	6.3	174	13.2	8.54	220	4.5	61	14	3.61	58	11	80	89.01	78.63			
4	GI	10.7	23.0	635	18.8	18.25	470	7.8	107	53	4.39	150	31	286	88.93	88.51			
5	ASICTA	9.8	18.7	516	1.6	1.63	42	4.3	59	0	0.71	0	0	0	0.00	0.00			
6	FLUKST	8.3	3.2	89	6.8	11.72	302	20.9	287	0	1.05	0	0	0	0.00	0.00			
7	STBED	4.2	4.2	133	←	←	←	←	←	←	←	←	←	←	←	←			
8	MACRO4	3.3	0.9	26	2.4	0.97	25	1.1	15	5	1.67	19	18	20	96.87	60.00			
9	MACRO2	3.0	0.7	19	2.3	0.73	19	0.2	3	3	6.33	280	20	540	99.97	100.0			
10	EUCLID	2.5	1.1	31	2.7	2.25	58	2.0	28	2	2.07	452	438	500	57.08	55.17			
11	SPATA	2.2	6.0	167	0.5	0.93	24	1.6	22	0	1.09	0	0	0	0.00	0.00			
12	BEARAG	1.9	0.4	11	2.1	1.59	41	2.0	27	9	1.52	255	7	500	51.84	56.09			
13	RESTOR	1.9	1.2	34	1.5	0.42	11	0.5	7	4	1.57	4029	20	11988	95.92	72.72			
14	APMATX	0.8	0.3	7	1.4	1.20	31	1.1	15	0	2.07	16084	11988	247645	88.67	51.61			
15	MACRO1	0.6	1.4	39	0.6	0.35	9	0.5	7	2	1.29	20	20	20	52.97	44.44			
16	FRSSOC	0.5	0.5	13	0.4	1.47	38	2.4	33	0	1.15	231820	231820	231820	99.33	13.15			
17	JMPUT	0.4	0.2	5	0.3	0.50	13	0.7	10	2	1.30	60	8	180	45.42	38.46			
					0.3	0.15	4	0.1	1	0	4.00	88933	88933	88933	83.73	75.00			
					1.8	3.03	78	5.5	76	0	1.02	0	0	0	0.00	0.00			
					0.3	0.35	9	0.1	1	0	9.00	65	65	65	99.56	88.88			
合計		97.8	90.9	2515	98.4	97.37	2508	97.9	1342	194	1.87					54.23			
参考				2766			2575		1370	194	1.88					54.33			
*印の合計										158						54.23			

(注) 参考欄の値は、順序で動かした組合のCPU時間等を示す。

Table 7.2 Speedup ratio by vector processing of MORSE code on FACOM VP2600.

	オリジナル版		オリジナルのベクトル版								修正したベクトル版								倍率 (OVV/ AVV)
	CPU (ms)	RGN (KB)	スカラモード				ベクトルモード (OVV)				スカラモード				ベクトルモード (AVV)				
			CPU (ms)	RGN (KB)	ERGN (KB)	VU時間 (ms)	CPU (ms)	RGN (KB)	ERGN (KB)	VU時間 (ms)	CPU (ms)	RGN (KB)	ERGN (KB)	VU時間 (ms)	CPU (ms)	RGN (KB)	ERGN (KB)	VU時間 (ms)	
入力データ1 (FNS)	2766	4360	2575	5660		1370	194	5852		2765	5724		1293	206	5920		1.06		
入力データ2 (FNSN)	3119	4396	3251	704	16040	1465	155	840	16432	3320	712	16624	1398	171	852	16684	1.05		
入力データ3 (LMFBR)	5844	4360	8431	5672		5076	1616	5788		8830	5724		4635	1608	5856		1.09		
入力データ4 (LMFBRN)	6033	4396	11996	708	16156	4222	529	764	16212	11014	712	16624	3654	522	788	16684	1.16		

入力データ1:中性子数 500 * 世代数 10
 入力データ2:中性子数 5000 * 世代数 1
 入力データ3:中性子数 500 * 世代数 100
 入力データ4:中性子数 5000 * 世代数 10

Table 7.3 Simulation results of vector-parallel processing of MORSE code on FACOM VP2600.

	オリジナル (d4)		ベクトルバージョン (d5)			
	スカラモード	スカラモード	ベクトル・パラレル (d6)			
			プロセッサ台数			
	1台	2台	3台	4台		
CPU時間 (d1)	25.900	27.650	12.930	8.782	7.766	7.002
倍率 1 (d2)	1.000	0.937	2.003	2.949	3.335	3.699
倍率 2 (d3)			1.000	1.472	1.665	1.847

* d2は, (d4のd1) / (d5の各d1)

* d3は, (d6の1台) / (d6の2~4台)

Table 7.5 Simulation results of vector-parallel processing of VIM code on FACOM VP2600.

	オリジナル (d4)		ベクトルバージョン (d5)			
	スカラモード	スカラモード	ベクトル・パラレル (d6)			
			プロセッサ台数			
	1台	2台	3台	4台		
CPU時間 (d1)	28.890	53.050	18.530	10.642	8.050	6.754
倍率 1 (d2)	1.000	0.545	1.559	2.714	3.589	4.277
倍率 2 (d3)			1.000	1.741	2.302	2.744

* d2は, (d4のd1) / (d5の各d1)

* d3は, (d6の1台) / (d6の2~4台)

Table 7.4 CPU distribution of VIM code.

オリジナル版						ベクトル化版									
FORUNE 順位	サブルー チン名	FORUNE コスト(L)	CPU (%)	CPU 時間 (**)	FORUNE コスト(L)	スカラモード		ベクトルモード		倍率	ベクトル長			ベクトル化率	
						CPU (%)	CPU 時間 (**)	CPU (%)	CPU 時間 (**)		平均	最短	最長	FORUNE	実測
1	CROSSI	43.9	43.9	12683	49.1	50.4	21243	39.0	7389	2.87	85	19	148	74.28	70.77
2	FLUKKI	14.4	10.3	2969	9.8	11.0	4633	23.4	4500	1.03	0	0	0	0.00	0.00
3	REACTI	11.6	13.4	3869	0.1	0.1	22	0.1	19	1.16	0	0	0	0.00	0.00
4	GGONE	11.6	12.5	3610	9.2	9.9	4172	11.5	2184	1.91	115	0	228	74.33	55.08
5	ELASTI	7.4	5.5	1578	0.5	0.5	230	1.4	256	0.90	0	0	0	0.00	0.00
6	GGONE	3.5	2.8	799	10.6	8.1	3428	7.5	1423	2.41	142	16	900	62.76	60.07
7	UNRESI	3.1	3.2	911	3.3	2.9	1210	2.1	403	3.00	76	11	130	83.08	80.74
8	RUTING	2.8	2.7	768	2.0	2.5	1036	4.8	908	1.14	453	6	800	0.19	12.26
9	PRINT	0.5	0.2	47	0.3	0.1	46	0.2	45	1.02	11	3	70	5.89	2.17
10	BIRTH	0.3	0.3	101	0.2	0.1	46	0.2	41	1.12	10	8	12	81.22	54.35
11	BECET	0.2	0.3	80	0.1	0.2	80	0.4	85	0.94	301	7	888	33.41	18.75
12	POINTE	0.1	0.0	7											
合計		99.4	94.9	27422	99.6	100.8	42523	100.3	18986	2.24					61.65
参考				28890			42190		18940	2.23					61.22
*月の合計							2680								61.63

(注) 参考欄の値は、単独で動かしただけの場合のCPU時間等を示す。

Table 7.6 CPU distribution of MCNP code.
M C N P コー ド C P U 時間測定一覽表

オ リ ジ ナ ル 版				ベ ク ト ル 化 版													
FORT UNE 順 位	#7 名	FORT UNE コ ス ト (%)	CPU 時 間 (ms)	FORT UNE 順 位	#7 名	FORT UNE コ ス ト (%)	ス カ ラー CPU コ ス ト (%)	ス カ ラー CPU 時 間 (ms)	CPU コ ス ト (%)	CPU 時 間 (ms)	VU 時 間 (ms)	倍 率	平 均	最 短	最 長	FORTUNE NE (%)	化 率 (%)
1	CHKCEL	25.5	1877	2	CHKCL1	21.5	18.4	4615	16.0	1799	1504	2.56	38	1	25095	2993.61	
2	TRACK	23.3	3008	1	TRACK1	29.3	28.4	7106	25.4	2865	1490	2.48	63	1	25087	3080.65	
3	LGEVAL	8.7	1055	6	LGEVAL	4.9	4.3	1069	9.6	1077	0	0.99				0.0	0.0
4	ACETOT	8.1	1245	3	ACETT1	8.1	9.2	2302	9.8	1109	479	2.08	63	1	25079	8672.63	
5	TALLY	7.4	960	5	TALLY	6.2	7.0	1746	10.1	1138	140	1.53	110	1	25051	3742.84	
6	HSTORY	7.0	1013	8	HSTORY	3.4	3.6	895	2.0	227	54	3.94				60.04	80.67
7	TALLYD	4.0	901	4	TALLYD	7.1	8.9	2225	3.0	338	80	6.58	79	1	25087	5788.40	
8	TALSHF	3.4	576	7	TALSHF	4.4	2.4	599	0.6	71	65	8.44	12	1	6999	9299.0	
9	TRANSM	1.5	241	9	TRNSM1	1.9	1.9	466	0.6	66	66	7.06	79	1	25098	70100.0	
10	JBIN	1.5	519	11	JBIN2	1.6	1.6	398	0.6	65	58	6.12				99.60	98.24
11	RANG	1.3	測定不能	19	RANG	0.5	←	測定不能								0.0	0.0
				21	RANG1	0.3	0.6	147	1.3	151	0	0.97				0.0	0.0
12	NEWCEL	1.1	311	10	NEWCL1	1.9	1.8	441	2.9	326	28	1.35	54	1	25023	8132.43	
13	COLIDN	1.0	303	14	COLID1	0.9	1.2	293	2.3	258	18	1.14	35	1	12638	9518.09	
14	CALCPS	0.8	145	15	CALCPI	0.7	0.8	202	0.3	38	32	5.32	55	1	13299	5097.03	
15	ACECOS	0.7	116	20	ACECOS	0.4	0.4	112	0.9	106	0	1.06				0.0	0.0
16	SURFAC	0.7	142	17	SURFAC	0.6	0.8	191	0.9	97	20	1.97	82	1	25064	7159.69	
17	ACECOL	0.6	133	18	ACECL1	0.6	1.5	375	2.2	247	31	1.52	35	1	12651	9842.40	
18	ROTAS	0.5	112	22	ROTAS	0.3	0.4	93	0.8	90	0	1.03				0.0	0.0
19	STARTP	0.4	92	24	STARTP	0.2	0.3	70	0.5	61	11	1.15				25.00	28.57
20	ANGL	0.4	135	16	ANGL1	0.6	0.8	207	1.7	191	71	1.08	8	1	25051	7242.03	
21	DDDET	0.4	121	23	DDDET	0.2	0.3	66	0.2	21	5	3.14	79	1	25084	5675.67	
28	TRNSPT	0.1	0										0	0			
				12	YCECAS	1.5	1.4	347	1.0	109	47	3.18				91.03	82.13
				13	YYSTAR	1.4	1.5	385	1.6	185	53	2.08				80.78	65.71
合計		98.4	13005			98.597	22435094	310635	4252	2.29						73.79	
参考			13690				25060			11280	4790	2.22				74.10	

Table 7.7 Performance in vector processing of MCNP code.

粒子数	スカラ処理時	ベクトル処理時		速度向上率	ベクトル化率
	CPU 時間	CPU 時間	VU時間		
250	315 秒	175 秒	61 秒	1.03 倍	64 %
500	318 秒	150 秒	49 秒	1.21 倍	68 %
1500	361 秒	127 秒	37 秒	1.43 倍	75 %
2500	376 秒	122 秒	35 秒	1.48 倍	77 %
オリジナル版CPU 時間		181 秒		1.0 倍	—

Table 7.8 Performance in vector processing of GRAPE code.

	original (opt2)	original (opt3)	optimized original	vectorized (scalar)	vectorized (vector)
cpu time	340sec.	373sec.	219sec.	274sec.	89sec.
speedup ratio	0.64	0.58	1.0	0.79	2.46

```

C          LGEVAL
C          FUNCTION LGEVAL(LG,N)
C          EVALUATE LOGICAL EXPRESSION LG OF LENGTH N.
C          REPRESENTATION OF LOGICAL ELEMENTS:
C          (      100001      TRUE      1
C          )      100002      FALSE     0
C          UNION 100003      INTERSECTION  IMPLICIT
C          DIMENSION LG(*)
C
C          LG(N+1)=100002
C          I=0
C          L=0
10  LGEVAL=1
20  I=I+1
    IF(LG(I).GT.1)GO TO 40
    LGEVAL=MIN(LGEVAL,LG(I))
    IF(LGEVAL.NE.0)GO TO 20
30  I=I+1
    IF(LG(I).LT.2)GO TO 30
40  IF(LG(I).NE.100003)GO TO 50
    IF(LGEVAL.EQ.0)GO TO 10
    IF(L.EQ.0)RETURN
    GO TO 60
50  IF(LG(I).NE.100001)GO TO 80
    L=L+1
    IF(LGEVAL.NE.0)GO TO 20
60  M=1
70  I=I+1
    IF(LG(I).EQ.100001)M=M+1
    IF(LG(I).EQ.100002)M=M-1
    IF(M.NE.0)GO TO 70
80  L=L-1
    IF(I.NE.N+1)GO TO 20
    RETURN
    END

```

Fig. 7.1 FORTRAN statements of subroutine LGEVAL in MCNP code.

8. おわりに

HASPの研究開発も5年を経過し、研究も幾分進展をみた。この間に研究所内外の多くの方々から貴重なコメント、助言を頂いたのは研究推進に役立っている。今後、蓄積していくソフトウェアの検証・評価のためには、ロボット・センサー、具体的な異動ロボット等のハードウェアが不可欠であり、外部機関との共同研究等新たな突破口も模索しなければならない。尚、本報告書の執筆担当者は、

第2章	神林 奨
第3章	藤井 実
第4章	久米 悦雄
第5, 6, 7章	樋口 健二
第1, 8章	秋元 正幸

である。また、全章に渡って内容のチェック等補筆訂正を海老原健一が行った。

謝 辞

HASPの研究開発に関し、中部大学西原宏教授、東京大学吉川弘之教授、九州大学松尾文顕教授からご批判、ご助言を頂いた。厚くお礼を申し上げます。

8. おわりに

HASPの研究開発も5年を経過し、研究も幾分進展をみた。この間に研究所内外の多くの方々から貴重なコメント、助言を頂いたのは研究推進に役立っている。今後、蓄積していくソフトウェアの検証・評価のためには、ロボット・センサー、具体的な異動ロボット等のハードウェアが不可欠であり、外部機関との共同研究等新たな突破口も模索しなければならない。尚、本報告書の執筆担当者は、

第2章	神林 奨
第3章	藤井 実
第4章	久米 悦雄
第5, 6, 7章	樋口 健二
第1, 8章	秋元 正幸

である。また、全章に渡って内容のチェック等補筆訂正を海老原健一が行った。

謝 辞

HASPの研究開発に関し、中部大学西原宏教授、東京大学吉川弘之教授、九州大学松尾文顕教授からご批判、ご助言を頂いた。厚くお礼を申し上げます。

付録A

1) BOXの場合

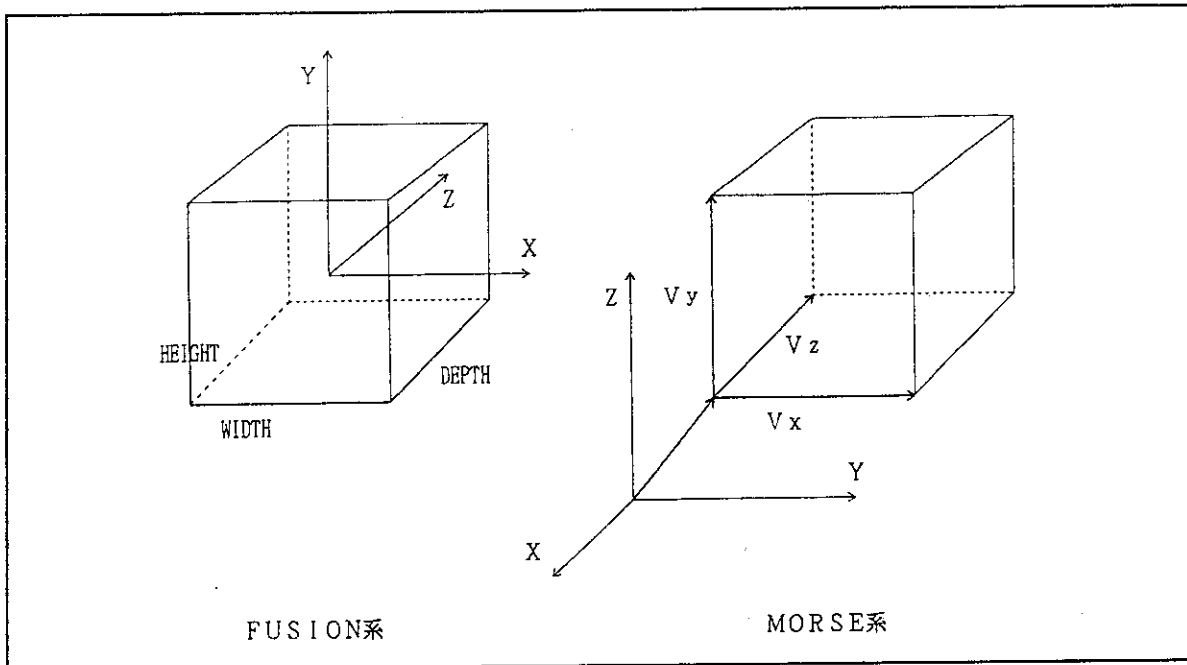


図3.1.1

① プリミティブ形状の変換

表3.1.1の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.1

FUSION系		MORSE系				備 考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
HEIGHT	H	Vx	H	0	0	ベクトル
WIDTH	W	Vy	0	W	0	ベクトル
DEPTH	D	Vz	0	0	D	ベクトル

② プリミティブの回転

MORSEでの各ベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

BOXの場合, 各ベクトルの $1/2$ のベクトルをFUSIONの物体中心位置から引くことによりFUSIONの中心をMORSEの物体中心に変換する。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に -1 をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

2) CONEの場合

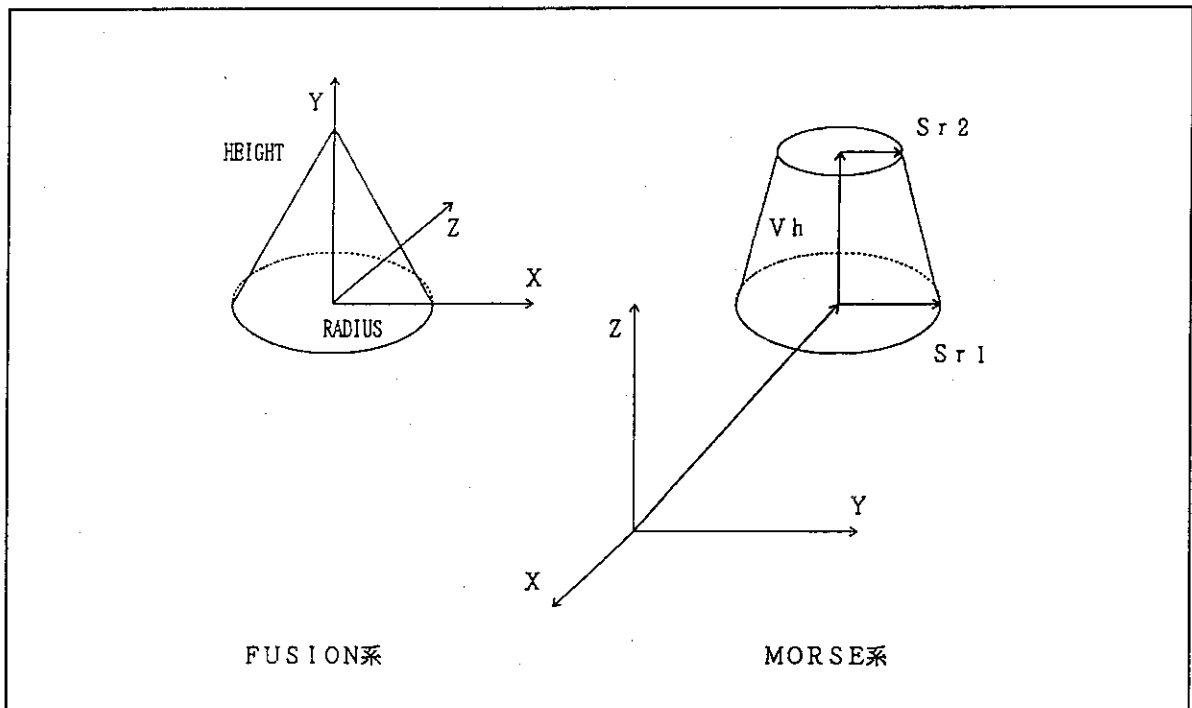


図3.1.2

① プリミティブ形状の変換

表3.1.2の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.2

FUSION系		MORSE系				備考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
HEIGHT	H	Vh	0	H	0	ベクトル
RADIUS	R	Sr 1	R			スカラー
—	—	Sr 2	0			スカラー

② プリミティブの回転

MORSE系のVhベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

CONEの場合, FUSIONの中心とMORSEの中心に同一である。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

3) CYLINDERの場合

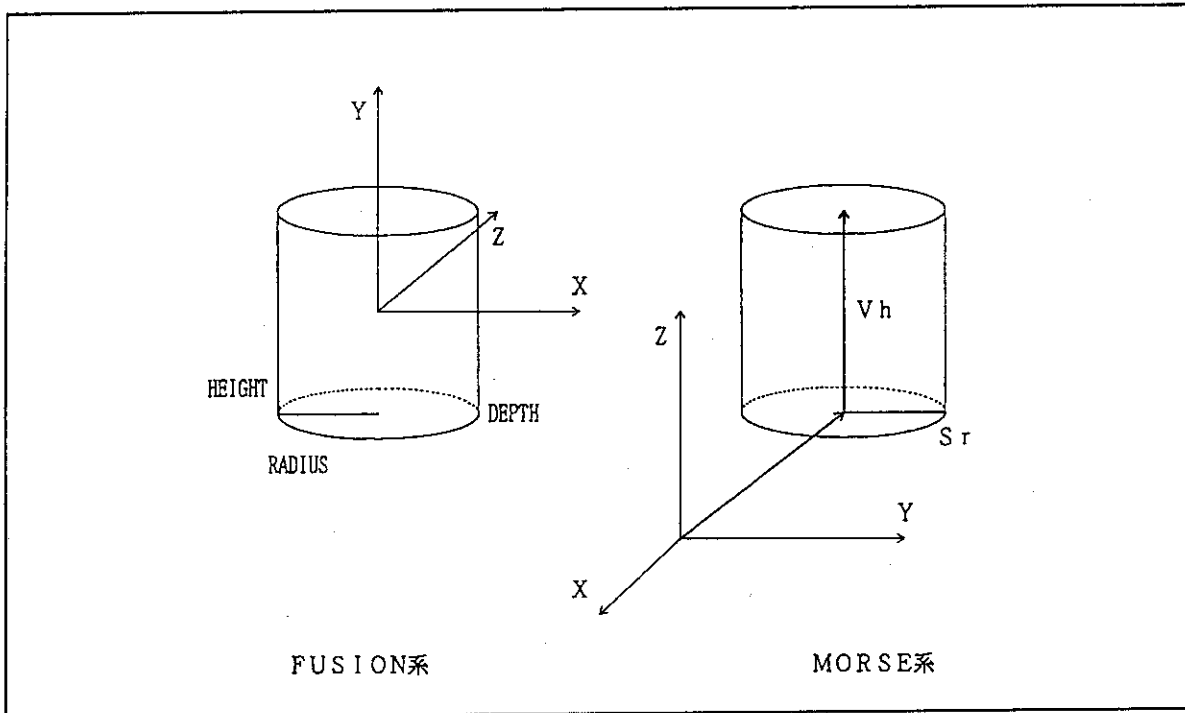


図3.1.3

① プリミティブ形状の変換

表3.1.3の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.3

FUSION系		MORSE系				備 考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
HEIGHT	H	Vh	0	H	0	ベクトル
RADIUS	R	Sr	R			スカラー

② プリミティブの回転

MORSE系のVhベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

CYLINDERの場合, Vhベクトルの1/2のベクトルをFUSIONの物体中心位置から引くことによりFUSIONの中心をMORSEの物体中心ベクトルに変換する。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

4) SPHEREの場合

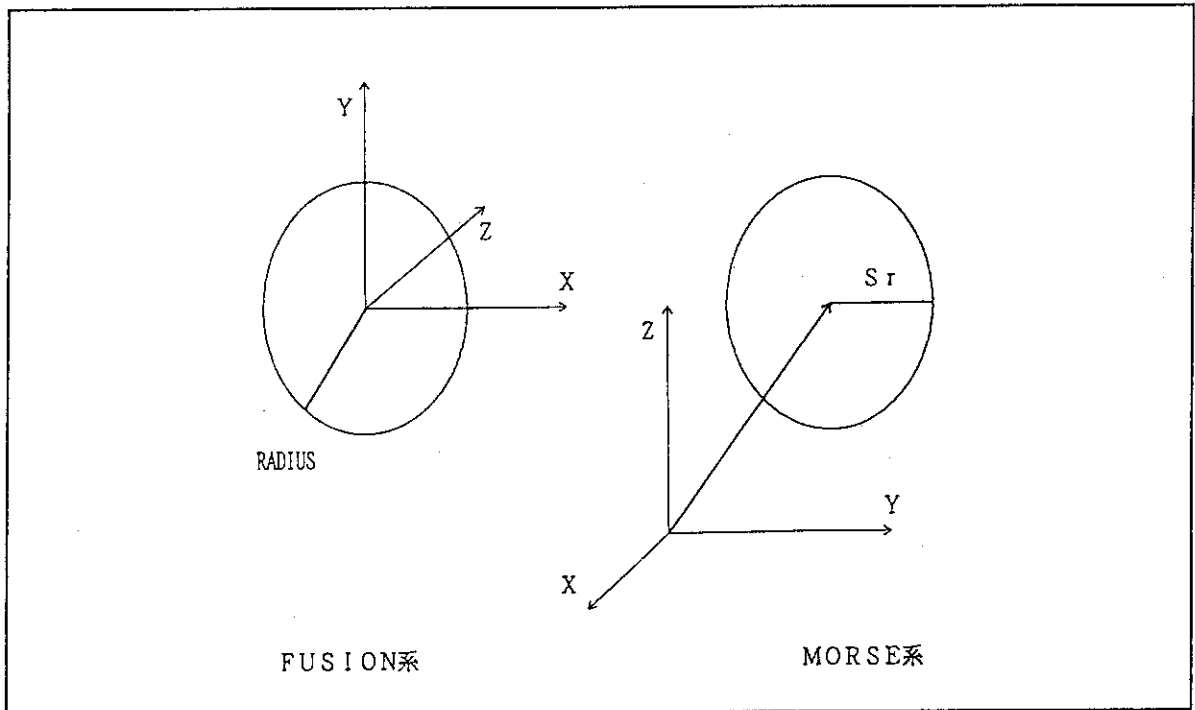


図3.1.4

① プリミティブ形状の変換

表3.1.4の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.4

FUSION系		MORSE系			備 考	
サイズ	値	ベクトル, スカラー名	X成分	Y成分		Z成分
RADIUS	R	Sr		R		スカラー

② プリミティブの回転

球なので回転しても, しなくても同じである。

③ プリミティブの位置の決定

SPHEREの場合, プリミティブの中心は変わらない。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

5) ELLIPSの場合

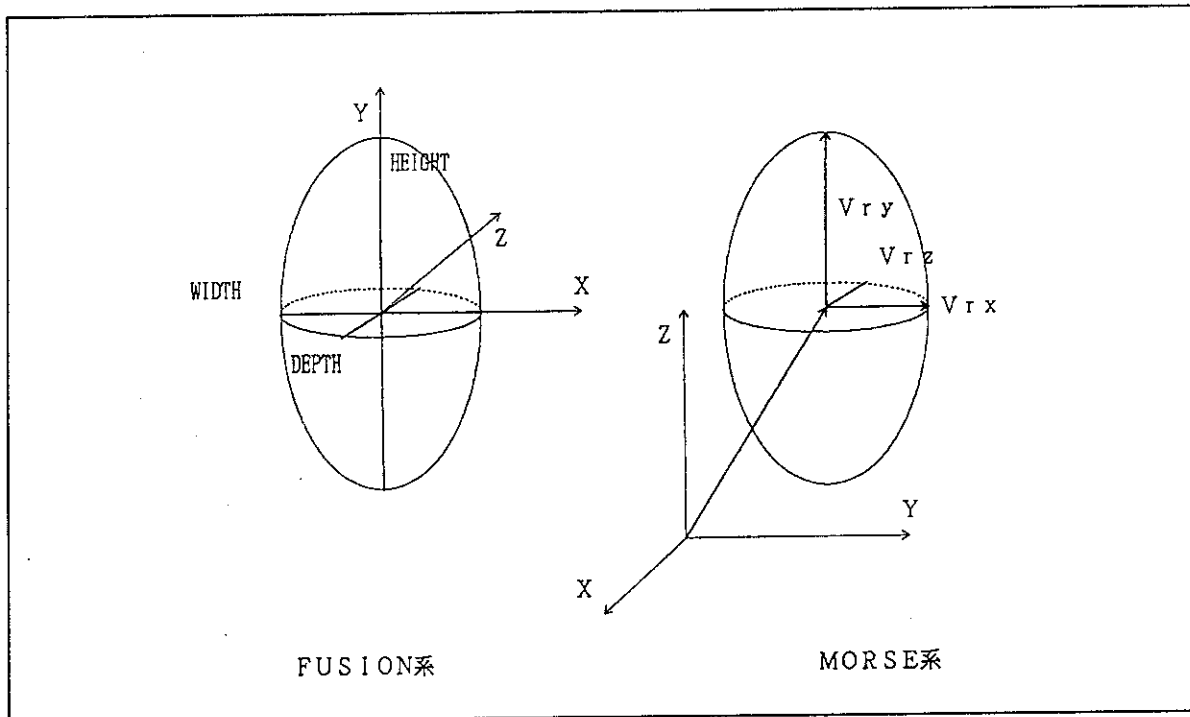


図3.1.5

① プリミティブ形状の変換

表3.1.5の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.5

FUSION系		MORSE系				備考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
WIDTH	W	Vrx	W	0	0	ベクトル
HEIGHT	H	Vry	0	H	0	ベクトル
DEPTHS	D	Vrz	0	0	D	ベクトル

② プリミティブの回転

MORSE系の各ベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

ELLIPSの場合, プリミティブの中心は変わらない。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

6) E-CYLINDERの場合

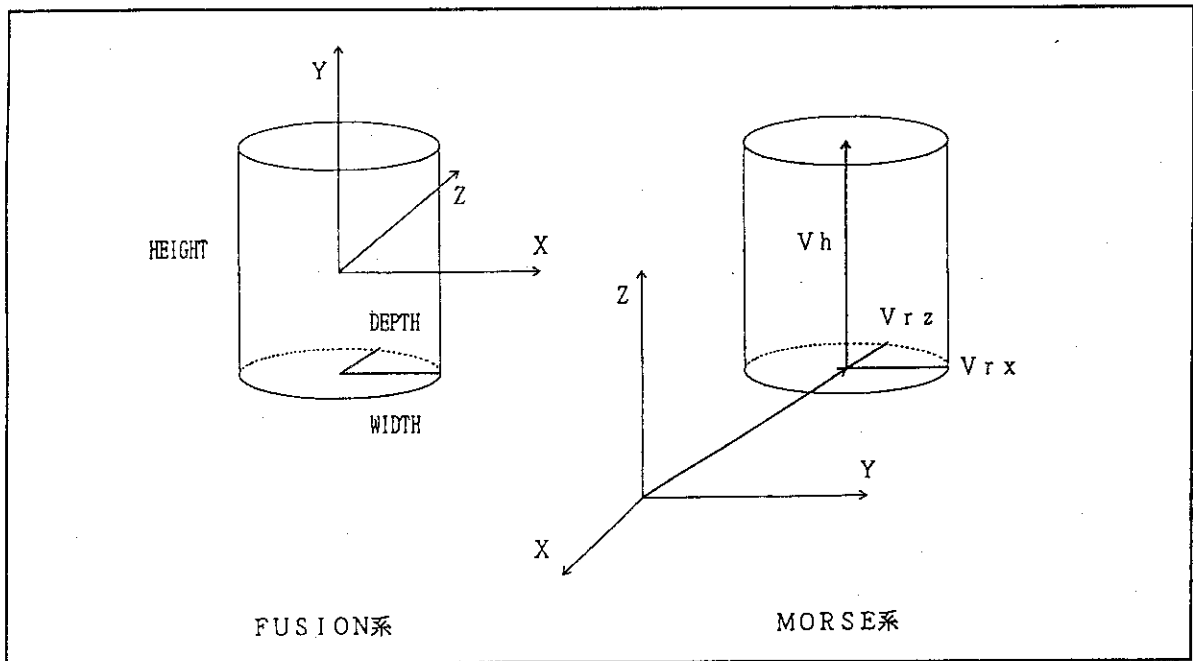


図3.1.6

① プリミティブ形状の変換

表3.1.6の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.6

FUSION系		MORSE系				備 考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
WIDTH	W	Vrx	W	0	0	ベクトル
HEIGHT	H	Vh	0	H	0	ベクトル
DEPTHS	D	Vrz	0	0	D	ベクトル

② プリミティブの回転

MORSEでの各ベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

E-CYLINDERの場合, Vhベクトルの1/2のベクトルをFUSIONの物体中心位置から引くことによりFUSIONの中心をMORSEの物体中心に変換する。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

7) HYPERの場合

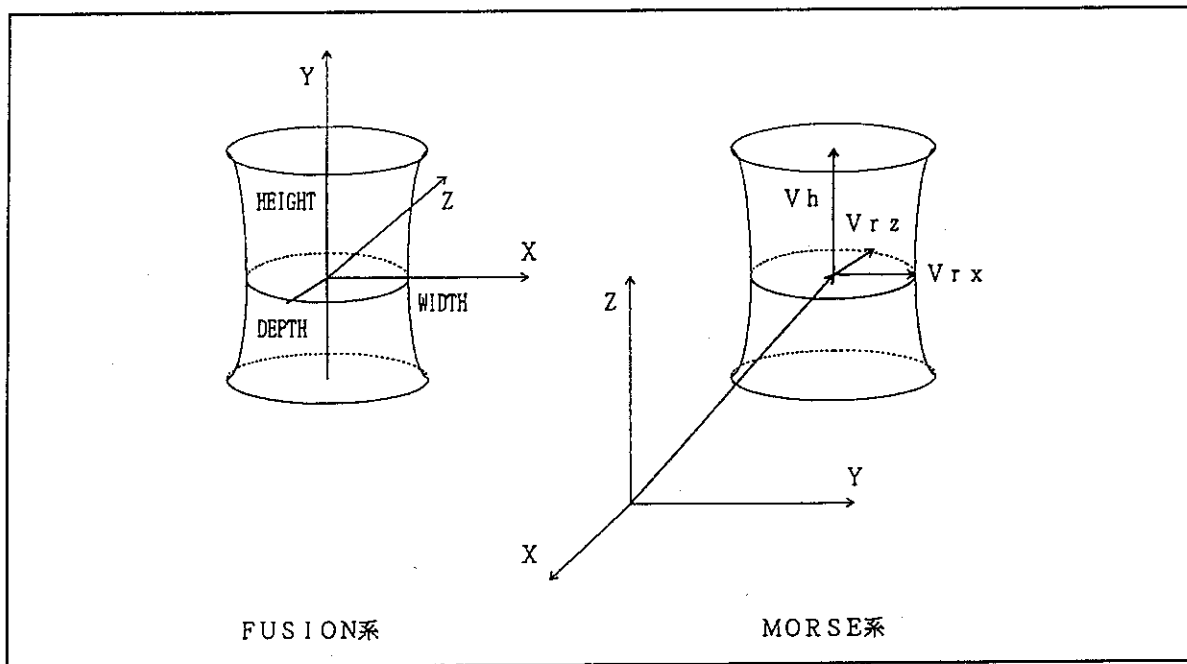


図3.1.7

① プリミティブ形状の変換

表3.1.7の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.7

FUSION系		MORSE系				備考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
WIDTH	W	Vrx	W	0	0	ベクトル
HEIGHT	H	Vh	0	H	0	ベクトル
DEPTHS	D	Vrz	0	0	D	ベクトル
RATIO	R	Sr	R			スカラー

② プリミティブの回転

MORSEでの各ベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

HYPERの場合, FUSIONの物体中心とMORSEの物体中心は同じである。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

8) PARABOLAの場合

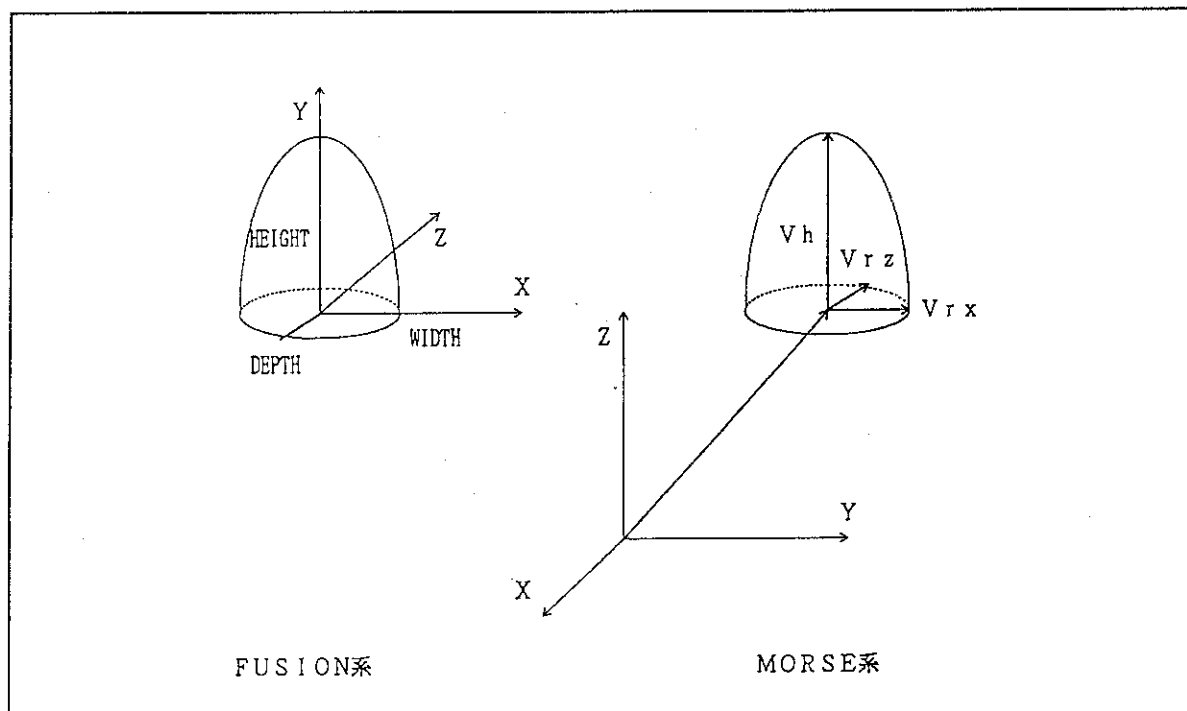


図3.1.8

① プリミティブ形状の変換

表3.1.8の様な対応でFUSIONの各サイズをMORSEのベクトル,もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.8

FUSION系		MORSE系				備考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
WIDTH	W	Vrx	W	0	0	ベクトル
HEIGHT	H	Vh	0	H	0	ベクトル
DEPTHS	D	Vrz	0	0	D	ベクトル

② プリミティブの回転

MORSEでの各ベクトルにFUSIONの回転ベクトルをかけることで,プリミティブを回転させる。

③ プリミティブの位置の決定

HYPERの場合, FUSIONの物体中心とMORSEの物体中心は同じである。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるなのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

9) TORUSの場合

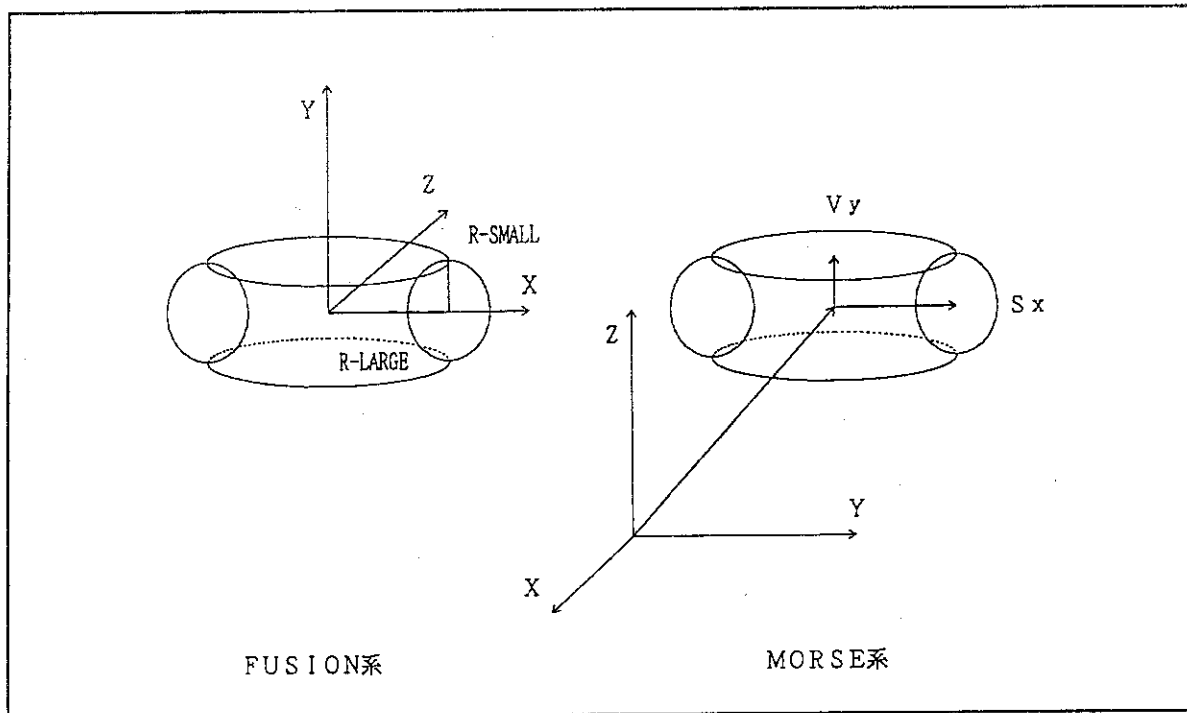


図3.1.9

① プリミティブ形状の変換

表3.1.9の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.9

FUSION系		MORSE系				備 考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
R-SMALL	R1	Vy	0	R1	0	ベクトル
R-LARGE	R2	Sx		R2		スカラー

② プリミティブの回転

MORSEでのVyベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

HYPERの場合, FUSIONの物体中心とMORSEの物体中心は同じである。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

10) ELLIPTICAL TORUSの場合

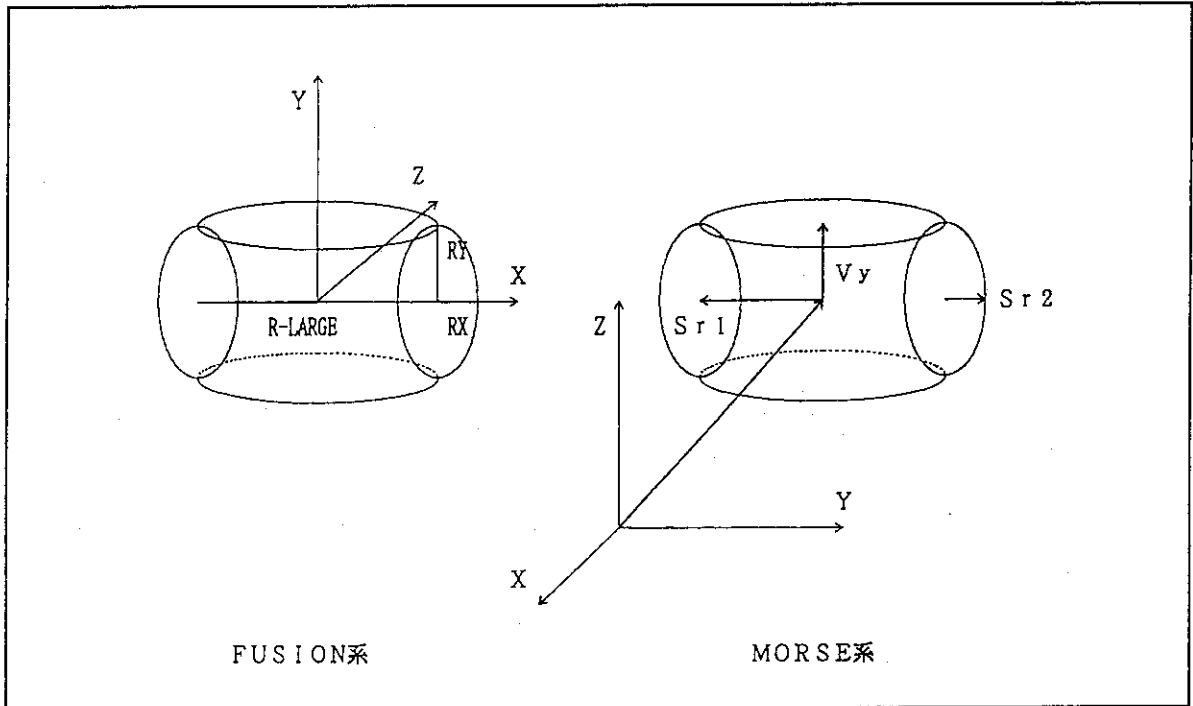


図3.1.10

① プリミティブ形状の変換

表3.1.10の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.10

FUSION系		MORSE系				備 考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
RY	RY	Vy	0	RY	0	ベクトル
RX	RX	Sr2		RX		スカラー
R-LARGE	RL	Sr1		RL		スカラー

② プリミティブの回転

MORSEでのVyベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

HYPERの場合, FUSIONの物体中心とMORSEの物体中心は同じである。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。

11) QUARTER TORUSの場合

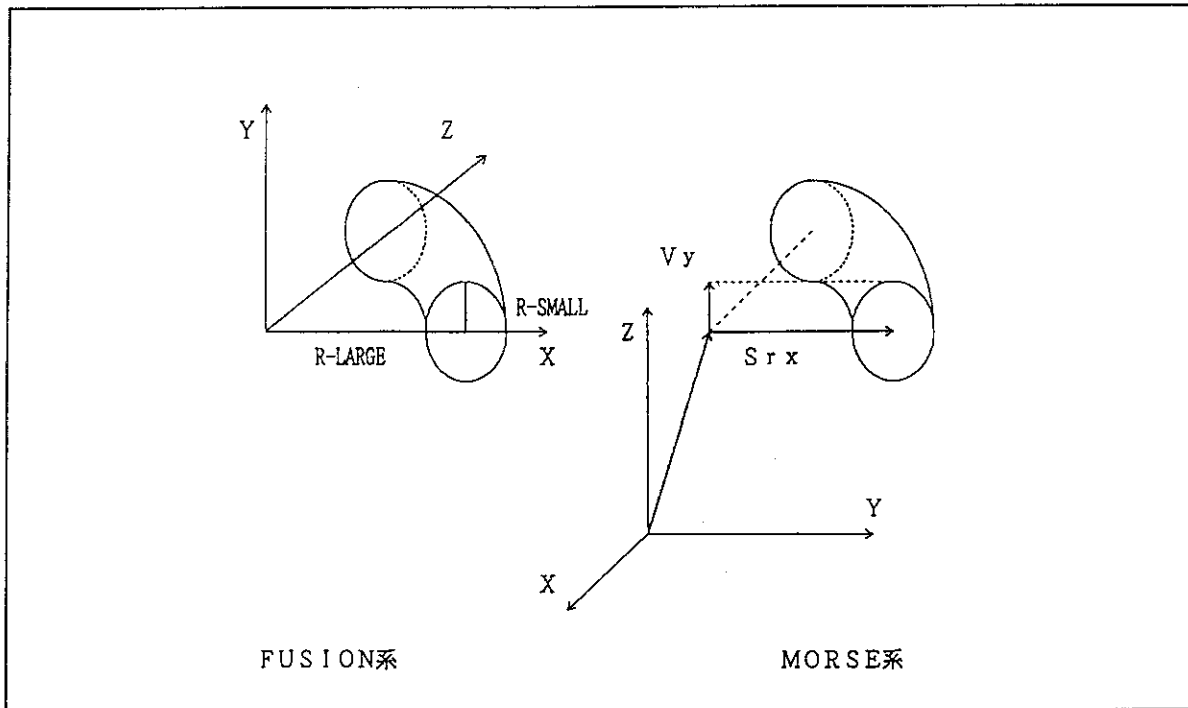


図3.1.11

① プリミティブ形状の変換

表3.1.11の様な対応でFUSIONの各サイズをMORSEのベクトル, もしくはスカラーに変換する。ここでMORSEのベクトルはFUSION座標系のベクトルである。

表3.1.11

FUSION系		MORSE系				備 考
サイズ	値	ベクトル, スカラー名	X成分	Y成分	Z成分	
R-SAMLL	R1	Vy	0	R1	0	ベクトル
R-LARGE	R2	Srx	R2			スカラー

② プリミティブの回転

MORSEでのVyベクトルにFUSIONの回転ベクトルをかけることで, プリミティブを回転させる。

③ プリミティブの位置の決定

Q-TORUSの場合, FUSIONの物体中心とMORSEの物体中心は同じである。

④ 座標系の変換

以上の3つの手順はFUSION系で行われたものであるのでMORSE系に変換しなければならない。MORSEの各ベクトルの成分をX成分をY成分にY成分をZ成分に, Z成分に-1をかけてX成分へ変換する。

以上の, 4つの手順で変換を行う。